SIEMENS
Ingenuity for life

Beispiele zum Webserver der SIMATIC S7-1200 / S7-1500

STEP 7 Basic (TIA Portal),
STEP 7 Professional (TIA Portal)

https://support.industry.siemens.com/cs/ww/de/view/68011496

Siemens Industry Online Support

Rechtliche Hinweise

Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG („Siemens“). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.
Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhaften Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.
Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.
Ergänzend gelten die Siemens Nutzungsbedingungen (https://support.industry.siemens.com).

Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.
Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.
Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.
Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter: https://www.siemens.com/industrialsecurity.
Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.
Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter: http://www.siemens.com/industrialsecurity.

© Siemens AG 2019 All rights reserved

Inhaltsverzeichnis

Rechtliche Hinweise **2**

1 Vorwort **5**

2 Verwendete Hard- und Software-Komponenten **6**

3 Grundlagen zu den Web-Standardseiten **8**

4 Grundlagen zu den Web-Anwenderseiten **9**

4.1 Login und Bedienungshinweise 9
4.2 Webserver - Initialisierung der Systemfunktion WWW 9
4.3 Bibliotheken - jQuery und S7-Framework 10

5 Lesen und Schreiben mit verschiedenen Variablentypen **12**

5.1 Automatisierungsaufgabe 12
5.2 Funktionsmechanismen und Bedienung 13
5.2.1 Aufbau des S7-Programms 15
5.2.2 Aufbau der Web-Anwenderseite (HTML-Datei) 16

6 Datum und Uhrzeit anzeigen **19**

6.1 Automatisierungsaufgabe 19
6.2 Funktionsmechanismen und Bedienung 19
6.2.1 Aufbau des S7-Programms 20
6.2.2 Aufbau der Web-Anwenderseite (HTML-Datei) 21

7 Anzeige eines S7-Arrays (nur S7-1500) **22**

7.1 Automatisierungsaufgabe 22
7.2 Funktionsmechanismen und Bedienung 22
7.2.1 Aufbau des S7-Programms 23
7.2.2 Aufbau der Web-Anwenderseite (HTML-Datei) 23

8 ENUM – Variablenwert durch Text ersetzt **24**

8.1 Automatisierungsaufgabe 24
8.2 Funktionsmechanismen und Bedienung 25
8.2.1 Aufbau des S7-Programms 26
8.2.2 Aufbau der Web-Anwenderseite (HTML-Datei) 26

9 HTTP-Redirection nach Fehler (nur S7-1500) **27**

9.1 Automatisierungsaufgabe 27
9.2 Funktionsmechanismen und Bedienung 28
9.2.1 Aufbau des S7-Programms 29
9.2.2 Aufbau der Web-Anwenderseite (HTML-Datei) 30

10 Sprachumschaltung auf Anwenderseiten **31**

10.1 Automatisierungsaufgabe 31
10.2 Funktionsmechanismen und Bedienung 31
10.2.1 Aufbau des S7-Programms 32
10.2.2 Struktur der Web-Anwenderseite 32
10.3 Erweiterung des Beispiels 37

11 Datenübertragung ohne wiederholten Seitenaufbau mit AJAX **38**

11.1 Automatisierungsaufgabe 39
11.2 Funktionsmechanismen und Bedienung 40
11.2.1 Aufbau des S7-Programms 41
11.2.2 Struktur der Web-Anwenderseite 42

© Siemens AG 2019 All rights reserved

12 Graph aus Daten eines Datalogs anzeigen 47

12.1 Automatisierungsaufgabe 47
12.2 Funktionsmechanismen und Bedienung 48
12.2.1 Aufbau des S7-Programms 49
12.2.2 Struktur der Web-Anwenderseite 50

13 Anzeigeelemente mit serieller Vektorgrafik 54

13.1 Programmieraufgabe 54
13.2 Funktionsmechanismen und Bedienung 54
13.2.1 Beispiel SVG-Element – Drehbewegung eines Motors 55

14 S7-Diagnose und Ladeindikator 58

14.1 Automatisierungsaufgabe 58
14.2 Funktionsmechanismen und Bedienung 58
14.2.1 Aufbau des S7-Programms 59
14.2.2 Struktur der Web-Anwenderseite 59

15 Wechseln von Bildern mit JavaScript 64

15.1 Programmieraufgabe 64
15.2 Funktionsmechanismen und Bedienung 64

16 Schaltfläche zum CPU Neustart 66

16.1 Automatisierungsaufgabe 66
16.2 Funktionsmechanismen und Bedienung 66
16.2.1 Aufbau des S7-Programms 67
16.2.2 Struktur der Web-Anwenderseite 67

17 Login auf Anwenderseiten 69

17.1 Automatisierungsaufgabe 69
17.2 Funktionsmechanismen und Bedienung 69
17.2.1 Aufbau des S7-Programms 70
17.2.2 Struktur der Anwenderseite 70

18 Performante Kommunikation via String 73

18.1 Automatisierungsaufgabe 73
18.2 Funktionsmechanismen und Bedienung 74
18.2.1 Aufbau des S7-Programms 75
18.2.2 Struktur der Web-Anwenderseite 78

19 Installation 82

19.1 Installation von Hard- und Software 82
19.2 Installation des Anwendungsbeispiels 83

20 Literaturhinweise 84

20.1 Literaturangaben 84
20.2 Internet-Link-Angaben 84

21 Historie 85

© Siemens AG 2019 All rights reserved

1 Vorwort

Ziel der Anwendungsbeispiele

Die Anwendungsbeispiele in diesem Dokument zeigen Ihnen Möglichkeiten, eigene Webseiten auf dem Webserver der S7-1200/1500 zu erweitern.

Dieses Dokument beschreibt die zum Download bereitgestellten Beispiele. Es empfiehlt es sich, die Beispiele auf einem Rechner ablaufen zu lassen und parallel mit Hilfe dieses Dokumentes zu analysieren.

Kerninhalte der Anwendungsbeispiele

Folgende Kernpunkte werden in den Anwendungsbeispielen behandelt:

- Lesen und Schreiben mit verschiedenen Variablentypen
- Uhrzeit anzeigen
- Arrays ausgeben (nur S7-1500)
- Anwenden des Datentyps ENUM
- HTTP-Redirection nach einem Fehler (nur S7-1500)
- Sprachumschaltung auf den Anwenderseiten
- Übertragen von Daten ohne erneuten Seitenaufbau
- Darstellung eines von der Steuerung erzeugten Datalogs als Graph
- Erstellen von Anzeigeelementen
- Anzeigen von Diagnoseinformationen
- Wechseln von Bildern mit JavaScript
- Neustarten der CPU über eine Schaltfläche
- Einloggen auf der Anwenderseite
- Performante Datenübertragung

© Siemens AG 2019 All rights reserved

Vorteile

Integrierter Webserver in der S7-1200 und S7-1500

Die Standard-Webseiten zum einfachen Anzeigen von Service und Diagnoseinformationen werden mit einem Klick aktiviert.

Zusätzlich können individuell gestaltete anwenderdefinierte Webseiten erstellt werden, die im Folgenden als „Anwenderseiten“ bezeichnet werden.

Ortsunabhängig

Die Abfrage der S7-Webseiten ist weltweit, über einen Internet Browser, möglich.

Anwendungsbeispiel

Universeller Einsatz des Anwendungsbeispiels für SIMATIC S7-1200 und S7-1500

Nutzen

Keine zusätzliche Hardware und Software erforderlich.

Der Zugriff auf den Webserver ist über große Entfernungen und mobile Kommunikationsgeräte wie z.B. Tablet PC, Smartphone usw. möglich.

Hinweis Die Anwendungsbeispiele in Verbindung mit dem Webserver sollen und können kein HMI System ersetzen.

2 Verwendete Hard- und Software-Komponenten

Die Anwendungsbeispiele wurden mit den folgenden Komponenten erstellt.

Hardware-Komponenten

Tabelle 2-1: Verwendete Hardware-Komponenten für die Beispiele

Komponente	Anz.	Bestellnummer	Hinweis
CPU 1511-1 PN	1	6ES7511-1AK01-0AB0	Firmware V2.6
CPU 1214C DC/DC/DC		6ES7214-1AG40-0XB0	Firmware V4.2
PG/PC mit Ethernet-Schnittstelle	1	-	-
IE FC TP STANDARD CABLE	1	6XV1840-2AH10	Verbindungsleitung IE Mindest-Bestellmenge 20m
RJ45 Steckverbinder	2	6GK1901-1BB10-2AA0	Konfektionierbar

Hinweis

Sie benötigen für diese Anwendungsbeispiele den aktuellen Firmwarestand der CPU. Abhängig vom Typ der CPU finden Sie in den folgenden Beiträgen weiterführende Links zu den entsprechenden Downloads:

- S7-1500: https://support.industry.siemens.com/cs/ww/de/view/78301349
- S7-1200: https://support.industry.siemens.com/cs/ww/de/view/107539750

© Siemens AG 2019 All rights reserved

Software-Komponenten

Tabelle 2-2: Verwendete Software-Komponenten für die Beispiele

Komponente	Anz.	Bestellnummer	Hinweis
SIMATIC STEP 7 Professional V15.1	1	6ES7822-1AA05-0YE5	-
Softwaretool zur Erstellung von HTML-Dateien, z. B. Frontpage, Notepad++, …	1	-	Webseiten erstellt mit Notepad ++
Webbrowser, z. B. Internet Explorer, Mozilla Firefox[1)]	1	-	Erstellt und getestet mit IE11 und Firefox

[1)] Folgende Webbrowser wurden für die Kommunikation mit der CPU getestet:

- Internet Explorer (Version 11)
- Mozilla Firefox (Version 50)

Hinweis

Die Anwendungsbeispiele sind für Firefox und den IE11 optimiert.
Für die Verwendung anderer Browser sind ggf. Anpassungen, bzgl. ihrer Darstellung im Browser vorzunehmen.

Gesamtaufbau

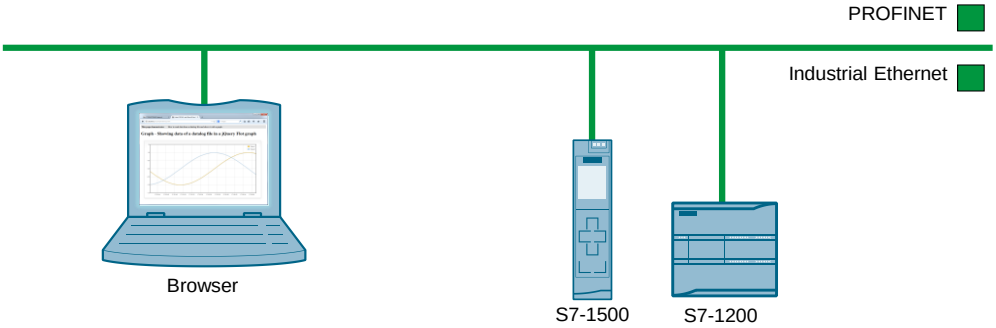
Die einzelnen Anwendungsbeispiele bestehen aus einem S7-Programm, das auf einer S7-1200/1500 abläuft und einer Webseite, die auf dem integrierten Webserver der S7 abläuft.

Ausnahme sind die Beispiele aus Kapitel 13 und 15, die allgemeine Web-Funktionen, ohne ein S7-Programm zeigen.

Die Webseiten werden mit einem Web-Browser aufgerufen und bestehen aus einer HTML-Datei als Basis.

Komplexere Beispiele beinhalten zusätzliche JavaScript-Dateien, mit der Dateiendung „js“.

Abbildung 2-1: Hardwareaufbau für die Anwendungsbeispiele.



Beispieldateien und Projekte

Die folgende Liste enthält die für dieses Beispiel erforderlichen Dateien, bestehend aus dem S7-Projekt und diesem PDF-Dokument.

Tabelle 2-3: Beispiele - Quelldateien

Komponente	Bemerkungen
68011496_examples_for_S7WebServer_CODE_v3.zip	Die Datei enthält das STEP 7 Projekt mit den jeweiligen HTML-Dateien im Verzeichnis \UserFiles\html.
68011496_examples_for_S7WebServer_de_DOC_v3.pdf	Dieses Dokument

© Siemens AG 2019 All rights reserved

3 Grundlagen zu den Web-Standardseiten

Voraussetzungen

In STEP 7 sind die folgenden Einstellungen der CPU Eigenschaften vorausgesetzt.

- Der Webserver muss aktiviert sein.
- Wenn Sie einen sicheren Zugriff auf die Webseiten benötigen, aktivieren Sie das Kontrollkästchen "Zugriff nur über HTTPS zulassen".
- Die automatische Aktualisierung der Webseiten ist aktiviert.
  Das Aktualisierungsintervall ist auf 10 Sekunden voreingestellt.
  Das Intervall kann im Bereich von 1s bis 999s eingestellt werden.

Zugriff über HTTP bzw. HTTPS

Mit der URL „http://ww.xx.yy.zz" bzw. „https://ww.xx.yy.zz" erhalten Sie Zugriff auf die Standard-Webseiten. Dabei entspricht „ww.xx.yy.zz" der IP-Adresse der CPU.

HTTPS dient zur Verschlüsselung und Authentifizierung der Kommunikation zwischen Browser und Webserver. Bei aktiviertem Kontrollkästchen „Zugriff nur über HTTPS zulassen" ist ein Aufruf der Webseiten der CPU nur über HTTPS möglich.

Zertifikat der CPU

Meldet der Browser einen Zertifikat Fehler ist wie im folgenden FAQ vorzugehen:

https://support.industry.siemens.com/cs/ww/de/view/63314183

Zugriffsrechte zum Anmelden auf den Webseiten unter STEP 7

Werksseitig ist in jeder SIMATIC S7-1200/1500 Steuerung der Benutzer "Everybody" ("Jeder") parametriert.

Dieser Benutzer hat eingeschränkte Zugriffsrechte und kein Passwort.
(Der Zugriff auf die Anwenderseiten ist dadurch unterbunden).

Für den vollen Zugriff auf die Anwenderseiten müssen Sie sich mit einem Benutzer einloggen, der die benötigten Zugriffsrechte besitzt. Benutzer, Passwort und Zugriffsrechte können Sie mit STEP 7 in den Eigenschaften der S7-1200/1500 Steuerung parametrieren.

Die Eingabefelder zum Anmelden finden Sie auf jeder Standard-Webseite der S7-1200/1500 Steuerung in der linken oberen Ecke.

Abbildung 3-1: Anmeldefenster zum Login auf der Standardwebseite

Standard-Webseiten von SIMATIC S7-1200 / S7-1500

Der Webserver der S7-1200 und S7-1500 bietet eine Vielzahl an Informationen zur jeweiligen CPU über die integrierte Standard-Webseite.

Der Aufbau der Standard-Webseiten ist im S7-1500 Webserver Funktionshand-buch detailliert beschrieben und nicht Gegenstand dieses Dokuments.

© Siemens AG 2019 All rights reserved

4 Grundlagen zu den Web-Anwenderseiten

Die Grundlagen zu den Anwenderseiten finden Sie in dem Anwendungsbeispiel „Eigene Webseiten für S7-1200/1500 erstellen und einsetzen“:

https://support.industry.siemens.com/cs/ww/de/view/68011496

4.1 Login und Bedienungshinweise

Im TIA Portal beginnt der jeweilige Name der Beispiele mit der Kapitelnummer:
Beispiel: Kapitel 5 entspricht dem Programmnamen „05ReadAndWriteS7...“

Die Web-Inhalte zu den Beispielen finden Sie im jeweiligen S7-Projekt auf Ihrem Rechner unter dem Ordner „…\html_...“.

Login der Anwenderseiten in den Applikationsbeispielen

User: **admin**
Passwort: **s7**

4.2 Webserver - Initialisierung der Systemfunktion WWW

In den Anwendungsbeispielen wird die Systemfunktion „WWW“ (SFC 99) durch den OB 1 aufgerufen.

Dieser Bausteinaufruf ist in allen Beispielen, mit entsprechendem Verweis auf dieses Kapitel, beschrieben.

Nr.	Funktion
1	Im OB1 wird die Systemfunktion „WWW“ (SFC 99) aufgerufen. Diese Funktion initialisiert den Webserver der CPU. Durch den zyklischen Aufruf der Funktion im OB1, ist eine Interaktion zwischen Webserver und Steuerung möglich. An die WWW-Funktion wird der Web-DB (DB 333) angeschlossen. Im Web-DB und in den Fragment-DB(s) ist der Aufbau der Anwenderseiten gespeichert.
2	Weitere Anweisungen und Bausteinaufrufe werden in den jeweiligen Anwendungsbeispielen beschrieben.

Hinweis

Grundlegende Informationen, zur Erzeugung der Datenbausteine für die Anwenderseiten finden Sie im folgendem Beitrag unter „Zum Einstieg“:

https://support.industry.siemens.com/cs/ww/de/view/68011496

© Siemens AG 2019 All rights reserved

4.3 Bibliotheken - jQuery und S7-Framework

Die Webseiten der Anwendungsbeispiele bestehen aus HTML-Dokumenten, JavaScripten und JavaScript-Bibliotheken.

Die Dateistrukturen finden Sie in den jeweiligen Anwendungsbeispielen.

Die JavaScript-Bibliotheken „jQuery“ und das S7-Framework werden in einigen Anwendungsbeispielen verwendet, die Sie in der nachfolgenden Tabelle finden.

Tabelle 4-1: Verwendung der Bibliotheken in folgenden Anwendungsbeispielen

Kapitel-Nr.	Kapitelname	jQuery	S7-Framework
10	Sprachumschaltung auf Anwenderseite	X	-
12	Graph aus Daten eines Datalogs anzeigen	X	X
14	S7-Diagnose und Ladeindikator	X	X
16	Schaltfläche zum CPU Neustart	X	X
17	Login auf Anwenderseite	X	X
18	Performante Kommunikation via String	X	X

In Ihrem Basis HTML-Dokument, binden Sie die jQuery-Bibliothek und die jeweiligen JavaScripte ein, die auf diese jQuery-Bibliothek zugreifen, bzw. diese nutzen. Das JavaScript „S7-Framework“ nutzt beispielsweise die jQuery-Bibliothek und steht im HTML-Code hier unterhalb des Aufrufs der jQuery-Bibliothek.

Abbildung 4-1: Aufruf von JavaScript-Bibliotheken im HTML-Dokument

```
<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>
    <script type="text/javascript" src="script/S7_framework_0.1.7.js"></script>
</head>
```

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei „jquery-2.1.3.min.js“ ist eine Bibliothek, die ein effizienteres Programmieren mit JavaScript ermöglicht. Die Bibliothek enthält grundlegende DOM-, Ereignis-, Effekt- und Ajax-Funktionen, auf die hier nicht im Einzelnen eingegangen wird. Typischerweise wird auf die Methoden der Bibliothek durch den vorangestellten Objektbezeichner „jQuery“ zugegriffen.
Dieser Objektbezeichner wird durch ein „$“-Zeichen ersetzt, was die Gesamtzahl an Zeichen in einem JavaScript verringert.

Beispiel:

jQuery.*post(URL, DATA)*

$.*post(URL, DATA)*

Allgemeine Informationen zu jQuery finden Sie unter: https://jquery.com/

© Siemens AG 2019 All rights reserved

S7_framework.js

Das JavaScript beinhaltet eine Reihe von Funktionen zur Konvertierung und Verarbeitung von Daten einer SIMATIC S7-1200/1500.

In der Programmierung des S7-Frameworks, werden Elemente aus der jQuery-Bibliothek verwendet. Daher müssen Sie das S7-Framework im HTML-Code stets nach der jQuery-Bibliothek aufrufen.

Die Anwendungsbeispiele bestehen jeweils getrennt für die S7-1200 und die S7-1500 Steuerungen. Beide Steuerungen greifen auf dieselben Webdateien zu.

Der Zugriff auf die Standard-Webserver unterscheidet sich jedoch CPU-abhängig durch die IDs, dem Login und weiterer Funktionen. Daher ist in den Beispielen eine Abfrage des Steuerungstyps realisiert.

Die Funktionen des S7-Frameworks werden in der nachfolgenden Tabelle aufgelistet.

Tabelle 4-2: Funktionen des S7-Frameworks

Nr.	Code-Beschreibung
1	Zuordnung der Variablentypen innerhalb des S7-Frameworks

	`var S7Framework = (function($, undefined){`
	`  // type of variable - 0=Bool, 1=unsigned INT, 2=signed INT, 3=real, 4=LReal, 5=String`
	`  var BOOL = 0, UINT = 1, INT = 2, REAL = 3, LREAL = 4, STRING = 5;`

2	PLC-Funktionen (Verschiedene Aufgaben als Bibliothek im Framework) - Funktionen und Varialblen für - AJAX (Datenübertragung) - JSON (Datenverarbeitung von Strings via Parser) - CPU Typ (Zugriffe auf Webserver S7-1200/1500) - Logon/Logoff (Referenziert auf S7 Standard-Webseite „Portal.mwsl“) - Lade Indikator (Definition und Funktion für animiertes Ladeicon) - Fehlerbehandlungen innerhalb der Datenverarbeitung des S7-Framework
	`function Controller () {`

Hinweis

Innerhalb des Codes finden Sie Informationen zu den Funktionen.
Eine Detailbeschreibung der Funktionen ist nicht Gegenstand dieser Anleitung.

© Siemens AG 2019 All rights reserved

5 Lesen und Schreiben mit verschiedenen Variablentypen

5.1 Automatisierungsaufgabe

Es sollen Variablen unterschiedlicher Datentypen gelesen und neu beschrieben werden.

Hinweis Der Datentyp DTL wird nur von der S7-1500 unterstützt.

Anforderungen an die Automatisierungsaufgabe

Es sind zwei HTML-Seiten zu programmieren.

- Programmieren einer HTML-Seite mit Variablen, verschiedenen Typs, gelesen und beschrieben werden.
- Programmieren einer HTML-Seite mit der Sondervariablen gelesen werden können.

© Siemens AG 2019 All rights reserved

5.2 Funktionsmechanismen und Bedienung

Gesamtaufbau

Das Applikationsbeispiel besteht aus einem S7-Programm und zwei HTML-Dokumenten, die über einen Browser als Webseiten angezeigt werden.

Aufbau der Anwenderseite

Die folgende Abbildung zeigt die Anwenderseite zur Anzeige einfacher PLC-Variablen, auf die lesend, oder schreibend zugegriffen wird.
Die oberen Variablen „Simple Variables“ greifen auf S7-Merker zu, wohingegen die Variablen „Structure und Array“ auf einen Datenbaustein „DB 1“ zugreifen.

Abbildung 5-1: Anwenderseite zum Lesen und Schreiben von einfachen Variablen

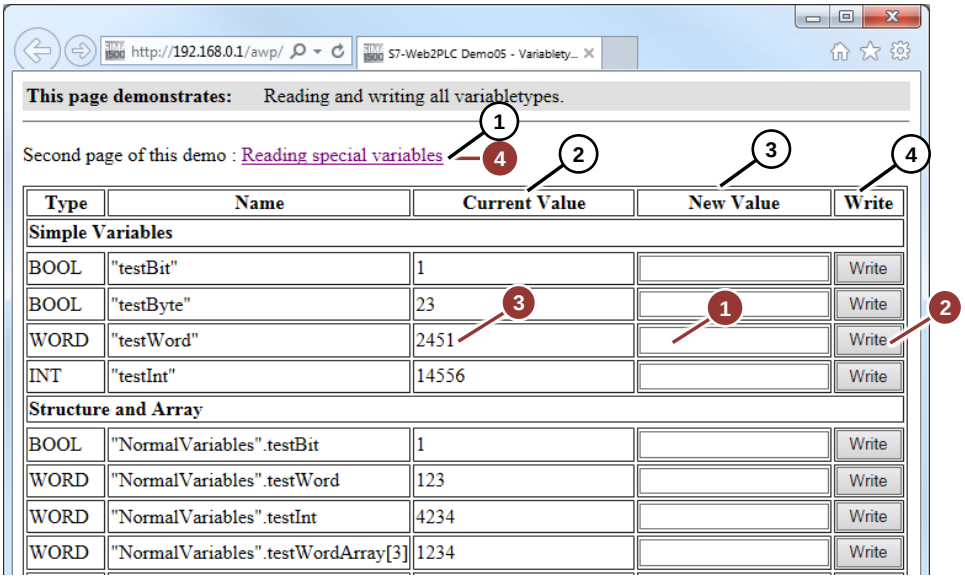


Tabelle 5-1: Webseite zum Lesen und Schreiben von einfachen Variablen

Position	Beschreibung
1	Dieser Link führt zu der HTML-Webseite „Reading special variables“ mit den Sondervariablen.
2	In dieser Spalte werden die aktuellen Werte der Variablen angezeigt.
3	In dieser Spalte wird der zu schreibende Wert eingetragen.
4	Mit den Schaltflächen „Write“ übertragen Sie die Werte an die CPU.

Tabelle 5-2: Handlungsanweisungen zur Bedienung der Anwenderseite

Schritt	Handlungsanweisung
1	Geben Sie einen Wert, entsprechend dem Datentyp, in der Spalte „New Value“ ein.
2	Klicken Sie die Schaltfläche „Write“.
3	Der Wert erscheint in der Spalte „Current Value“
4	Möchten Sie die Webseite „Reading special variables“ öffnen, klicken Sie auf diesen Link.

© Siemens AG 2019 All rights reserved

Die folgende Abbildung zeigt den Aufbau der Anwenderseite zum Lesen von Sondervariablen.

Abbildung 5-2: Anwenderseite zum Lesen und Schreiben von Sondervariablen

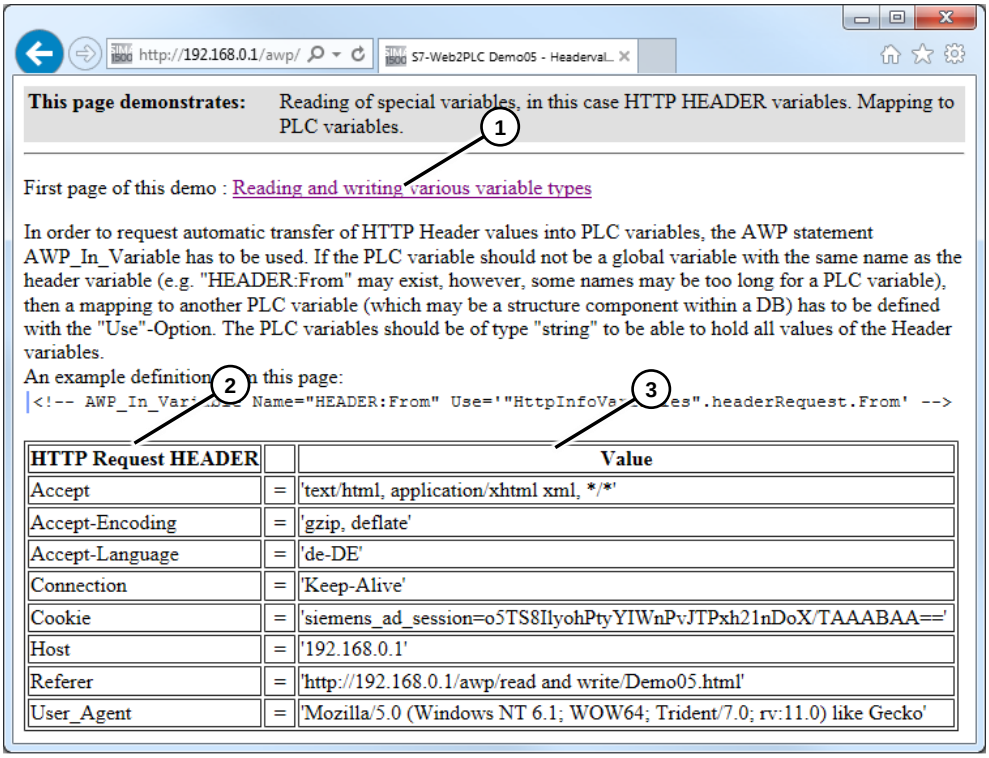


Tabelle 5-3: Webseite zum Lesen und Schreiben von Sondervariablen

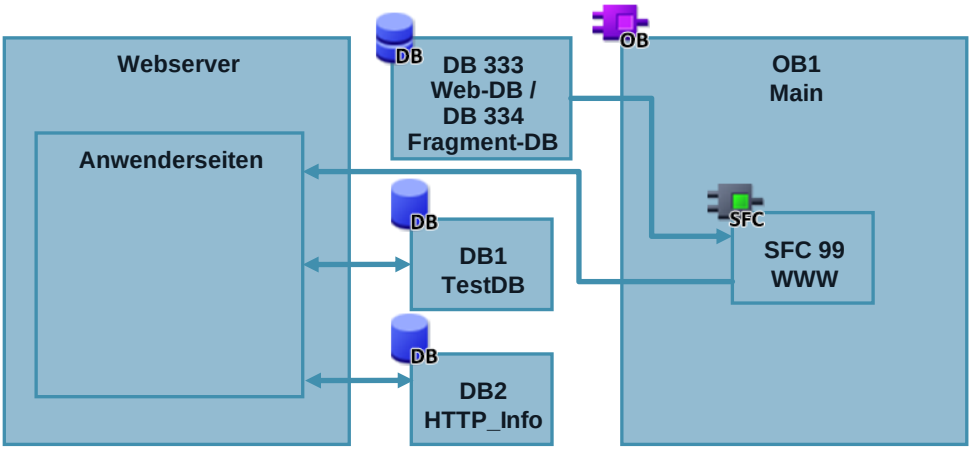
Position	Beschreibung
1	Dieser Link führt zur Webseite zum Lesen und Schreiben von PLC-Variablen
2	In dieser Spalte stehen die Namen der Sondervariablen
3	In dieser Spalte steht die Information (Wert) zur jeweiligen Sondervariable

© Siemens AG 2019 All rights reserved

5.2.1 Aufbau des S7-Programms

In der nachfolgenden Abbildung ist die S7-Programmstruktur schematisch dargestellt. Die HTML-Webseite wird zyklisch aufgerufen. Der Datenaustausch mit den Bausteinen wird nachfolgend beschrieben.

Abbildung 5-3: S7-Programm zum Lesen und Schreiben von Variablen



Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

Inhalt der PLC Variablen

In der ersten HTML Seite sind im oberen Bereich einfache PLC-Variablen (Merker) definiert, über die Webseite gelesen und beschrieben werden können.

Inhalt des DB1

Im „TestDB“ (DB1) sind Variablen definiert, welche über die Webseite gelesen und beschrieben werden können. Ausnahme ist die Variable „WWW_RET_VAL“. Diese Variable enthält den Rückgabewert der WWW-Funktion.

Inhalt des DB2

Im DB2 „HTTP_Info“ sind Sondervariablen, in diesem Fall HEADER_Request-Variablen, gespeichert.
Diese Variablen können vom User über die Webseite nur gelesen werden.
Beschrieben werden diese Variablen über den Webserver durch die Webseite.

© Siemens AG 2019 All rights reserved

5.2.2 Aufbau der Web-Anwenderseite (HTML-Datei)

Nr.	Code-Abschnitt

1	Um eine PLC-Variable aus den PLC-Variablen (tag table) von einer Webseite zu lesen und zu beschreiben ist nur der „Variablenname“ innerhalb des Syntax relevant.
	Allgemein :=“Variablenname“:
	
	Beispiel :=“testBit“:
	
	Default tag table
	Name / Data type
	1 testBit Bool
	2 testByte Byte
	3 testWord Word
	4 testInt Int
	
	`<!-- AWP_In_Variable Name='"testBit"' -->`
	
	`<tr>`
	`<form method="post" action="" >`
	`<td>BOOL</td><td>"testBit"</td><td>:="testBit":</td>`
	`<td><input type="text" name='"testBit"' maxlength="8" ></td>`
	`<td><input type="submit" value="Write"></td>`
	`</form>`
	`</tr>`

© Siemens AG 2019 All rights reserved

Nr.	Code-Abschnitt
2	Um eine Variable aus einem DB zu lesen und zu beschreiben, sind der DB-Name und der Variablenname relevant.

NormalVariables

	Name	Data type
1	Static	
2	testBit	Bool
3	testWord	Word
4	testInt	Int
5	testWordArray	Array[0..3] of Word
6	testBitArray	Array[0..3] of Bool
7	testChar	Char
8	testByte	Byte
9	testDWord	DWord
10	testDInt	DInt
11	testReal	Real
12	testString	String
13	testTime	Time
14	testDate	Date
15	testToD	Time_Of_Day
16	testDT	DTL

```
<!-- AWP_In_Variable Name='"NormalVariables".testBit' -->

<tr>
<form method="post" action="">
<td>BOOL</td>
<td>"NormalVariables".testBit</td>
<td>:="NormalVariables".testBit:</td>
<td>
    <input type="text" name='"NormalVariables".testBit' maxlength="8" >
</td>
<td>
    <input type="submit" value="Write">
</td>
</form>
</tr>
```

© Siemens AG 2019 All rights reserved

Nr.	Code-Abschnitt

3	Die Sondervariablen (hier: HTTP-Request-Variablen) werden mit AWP-Kommandos in den DB-Variablen abgespeichert.
	Diese DB-Variable wird von der HTML-Datei gelesen.
	
	HttpInfoVariables
	Name / Data type
	1 Static
	2 headerRequest "typeHeaderReques
	3 accept String
	4 acceptEncoding String
	5 acceptLanguage String
	6 connection String
	7 cookie String
	8 host String
	9 referer String
	10 userAgent String
	
	`<!-- AWP_In_Variable Name="HEADER:Accept"`
	`Use='"HttpInfoVariables".headerRequest.accept' -->`
	
	`:="HttpInfoVariables".headerRequest.accept:`
	
	Hinweis:
	http-Request- Variablen sind globale Variablen im Geltungsbereich eines Skriptes

© Siemens AG 2019 All rights reserved

6 Datum und Uhrzeit anzeigen

6.1 Automatisierungsaufgabe

Es soll die Uhrzeit der CPU auf einer Webseite angezeigt werden.

Anforderungen an die Automatisierungsaufgabe

- Erfassen der Uhrzeit im STEP 7-Programm mit der Funktion „RD_LOC_T“.
- Programmieren einer Webseite, auf der die Uhrzeit ausgegeben wird.
- Aktualisieren der Uhrzeit mit Hilfe einer in der HTML-Datei eingebundenen zweiten HTML-Datei (Inlineframe).
  Eine Beschreibung des Begriffs Inlineframe finden Sie im Kapitel 11.

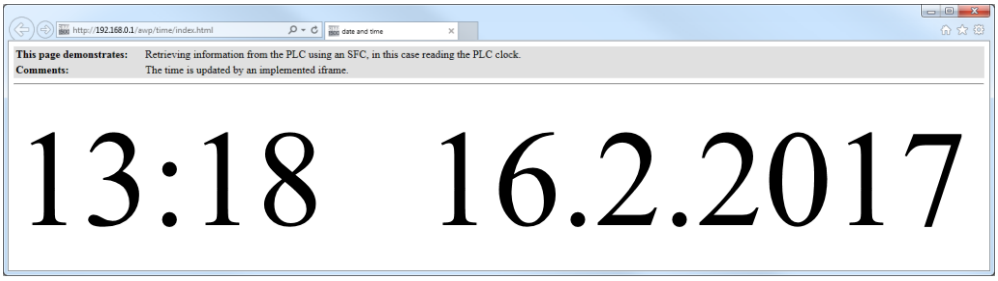
6.2 Funktionsmechanismen und Bedienung

Das Applikationsbeispiel besteht aus einem S7-Programm und zwei HTML-Dokumenten, die über einen Browser als Webseiten angezeigt werden.

Aufbau der Anwenderseite

Die Uhrzeit und das Datum werden in der Anwenderseite, horizontal und zentriert, angezeigt.

Abbildung 6-1: Anwenderseite zur Anzeige von Datum und Uhrzeit



Einstellen der Uhrzeit in der Steuerung

Im ersten Schritt ist die aktuelle Uhrzeit in der Datenquelle, also in der CPU einzustellen.

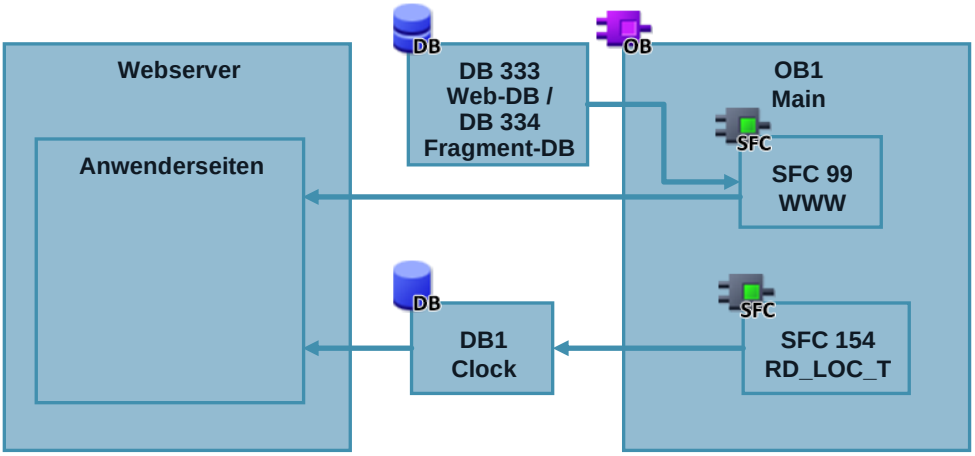
1. Öffnen Sie das TIA Portal und das Programmbeispiel.
2. Klicken Sie in der Projektnavigation mit der rechten Maustaste auf Ihre Steuerung und wählen Sie „Online verbinden“.
3. Wählen Sie „Online & Diagnose“.
4. Unter Funktion wählen Sie „Uhrzeit einstellen“.
5. In den Eigenschaften wählen Sie die Zeitzone und ggf. die Sommer und Winterzeit.

© Siemens AG 2019 All rights reserved

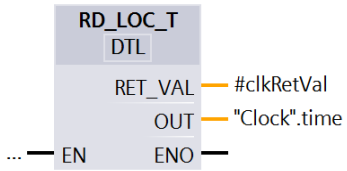
6.2.1 Aufbau des S7-Programms

In der nachfolgenden Abbildung ist die S7-Programmstruktur schematisch dargestellt. Die HTML-Anwenderseite wird zyklisch aufgerufen. Der Datenaustausch mit den Bausteinen wird nachfolgend beschrieben.

Abbildung 6-2: S7-Programm zur Anzeige von Datum und Uhrzeit



Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.
2	Die Anweisung „RD_LOC_T“ (SFC 154) liest die Uhrzeit der CPU und speichert diese in dem DB „Clock“ (DB 1) in der Variable „time“ ab. 

Inhalt des Clock-DB (DB1)

In diesem DB wird die Zeit im Format „DTL“ gespeichert.
Die Anwenderseite liest aus diesem DB die Uhrzeit aus.

© Siemens AG 2019 All rights reserved

6.2.2 Aufbau der Web-Anwenderseite (HTML-Datei)

Nr.	Code-Abschnitt

1	Durch die AWP-Kommandos werden die Variablen initialisiert.
	`<!-- AWP_In_Variable Name='"Clock".time.YEAR' -->`
	`<!-- AWP_In_Variable Name='"Clock".time.MONTH' -->`
	`<!-- AWP_In_Variable Name='"Clock".time.DAY' -->`
	`<!-- AWP_In_Variable Name='"Clock".time.HOUR' -->`
	`<!-- AWP_In_Variable Name='"Clock".time.MINUTE' -->`
	`<!-- AWP_In_Variable Name='"Clock".time.SECOND' -->`

2	In diese div-Box schreibt das Inlineframe („Update_Page.html“-Datei) die Uhrzeit.
	`<div id="clock">`
	`</div>`

3	Das Inlineframe muss in die „index.html“-Datei eingebunden werden, damit es im Funktionsablauf integriert ist. `<iframe src="Update_Page.html" style="display:none;"/>`

4	In der Datei „Update_Page.html“ wird angegeben, welche Variablen zyklisch aktualisiert werden und wo diese hinterlegt werden sollen.
	(Dies ist über die ID im JavaScript und HTML-Code umgesetzt.)
	`/* Function for updating the variables */`
	`function updateSingleVariablesTable()`
	`{`
	`var table = document.getElementById("singleVariablesTable");`
	`for(i = 1; i < table.rows.length; i++)`
	
	Referenziert auf:
	`<table id="singleVariablesTable" border="1">`

	Zum Anzeigen einer „0“, z. B. bei 12:03Uhr, ist die If-Abfrage vorangestellt.
	`<td><!-- read of date and time -->`
	`<script type="text/javascript">`
	`var hour = :="Clock".time.HOUR:;`
	`var minute = :="Clock".time.MINUTE:;`
	`if (hour < 10)`
	`{`
	`hour= "0" + hour;`
	`}`
	`if (minute < 10)`
	`{`
	`minute = "0" + minute;`
	`}`
	`document.write(hour + ":");`
	`document.write(minute);`
	`document.write('   :="Clock".time.DAY:.');`
	`document.write(':="Clock".time.MONTH:.');`
	`document.write(':="Clock".time.YEAR:');`
	`</script>`
	`</td>`
	`<td>`
	`clock`
	`</td>`
	
	Informationen zu dieser Aktualisierungsmethode finden Sie im FAQ:
	http://support.automation.siemens.com/WW/view/de/97044123

© Siemens AG 2019 All rights reserved

7 Anzeige eines S7-Arrays (nur S7-1500)

7.1 Automatisierungsaufgabe

Es sollen alle Felder eines S7-Arrays gelesen werden und in einer Tabellenform auf einer Anwenderseite ausgegeben werden.

Anforderungen an die Automatisierungsaufgabe

- Erstellen eines Arrays in einem DB des STEP 7 Programms.
- Anzeige der Felder des Arrays in einer Tabelle auf einer Anwenderseite.

7.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Auf der Anwenderseite werden die Werte des Arrays strukturiert in einer Tabelle angezeigt. Die Aktualisierung der Werte im Array erfolgt durch eine Aktualisierung der Anwenderseite über „F5“.

Abbildung 7-1: Anwenderseite zur Anzeige von S7-Arrays

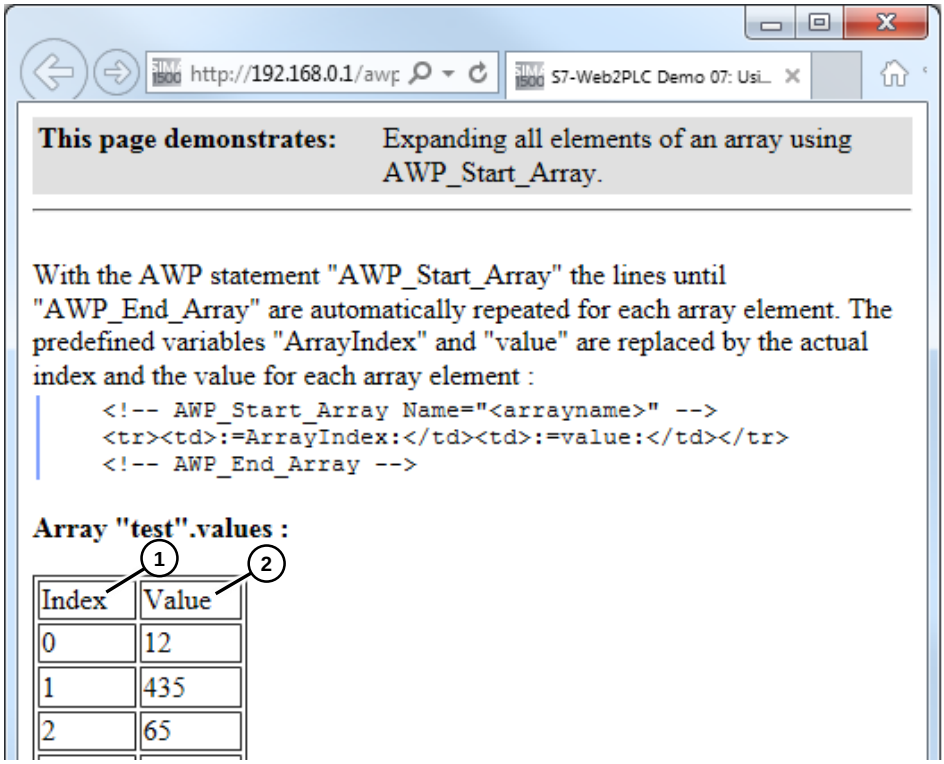


Tabelle 7-1: Anwenderseite zur Anzeige von S7-Arrays

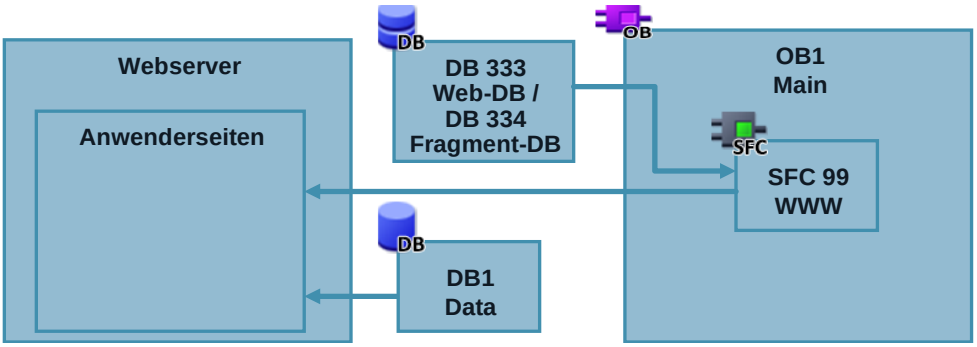
Position	Beschreibung
1	In dieser Spalte wird der Arrayindex angezeigt
2	In dieser Spalte wird der Wert des Arrayfeldes ausgegeben

© Siemens AG 2019 All rights reserved

7.2.1 Aufbau des S7-Programms

In der nachfolgenden Abbildung ist die S7-Programmstruktur schematisch dargestellt. Die Änderung von Werten des Arrays wird über die Onlineansicht des DB1 gesteuert und die Anwenderseite entsprechend Kapitel 7.2 aktualisiert.

Abbildung 7-2: S7-Programm zur Anzeige von S7-Arrays



Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

Inhalt des Data-DB (DB1)

Im Data DB1 ist das Array definiert, das von der Anwenderseite gelesen wird.

7.2.2 Aufbau der Web-Anwenderseite (HTML-Datei)

Das Array enthält die auf der Anwenderseite angezeigten Variablen.

Nr.	Funktion
1	Das AWP-Kommando müssen Sie auf den jeweiligen Arraynamen (DB) anpassen. Data Name / Data type / Start value 1 Static 2 values — Array[0..9] of Int 3 values[0] — Int — 12 4 values[1] — Int — 435 5 values[2] — Int — 65 6 values[3] — Int — 67 7 values[4] — Int — 234 8 values[5] — Int — 54 9 values[6] — Int — 214 10 values[7] — Int — 625 11 values[8] — Int — 824 12 values[9] — Int — 633 <table border="1" width="120px" > <tr><td>Index</td><td>Value</td></tr> <!-- AWP_Start_Array Name='"Data".values' --> <tr><td>:=ArrayIndex:</td><td>:=value:</td></tr> <!-- AWP_End_Array --> </table>

© Siemens AG 2019 All rights reserved

8 ENUM – Variablenwert durch Text ersetzt

8.1 Automatisierungsaufgabe

Der Integer Wert einer Variablen soll mit verschiedenen Schaltflächen auf einer Anwenderseite verändert werden.

Die Werte, die in die Variable geschrieben werden, sollen mit verschiedenen Meldetexten verknüpft werden.

In einer Textanzeige sollen die jeweiligen Meldetexte angezeigt werden.

Anforderungen an die Automatisierungsaufgabe

- Beschreiben und Lesen einer als ENUM definierten Variable mit einer Anwenderseite.
- Definieren einer ENUM-Variable.
- Erstellen einer Variable in einer PLC-Variablentabelle.

Hinweis

ENUM-Variablen werden in einer HTML-Datei definiert.
Bei ENUM-Variablen werden numerische Werte durch eine Zeichenkette ersetzt.
Im S7-Programm wird eine numerische Variable verwendet.

© Siemens AG 2019 All rights reserved

8.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Abbildung 8-1: Anwenderseite ENUM - Variablenwerte werden durch Text ersetzt

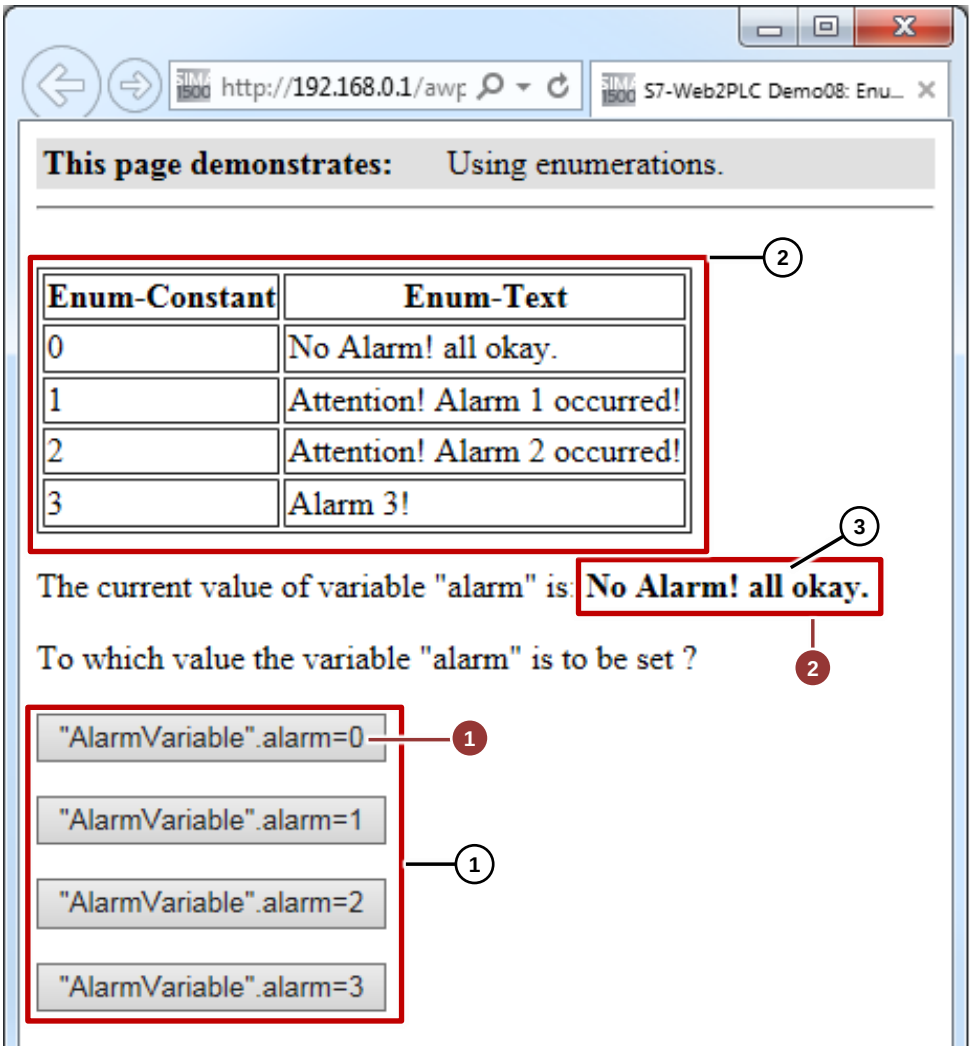


Tabelle 8-1 Anwenderseite ENUM - Variablenwerte werden durch Text ersetzt

Position	Beschreibung
1	Mit den markierten Schaltflächen kann die Variable „Alarm“ auf einen Wert zwischen 0 und 3 gesetzt werden.
2	Diese Tabelle zeigt die Zuordnung des Textes zum numerischen Wert.
3	Hier wird der entsprechende Text ausgegeben

Tabelle 8-2: Handlungsanweisungen zur Bedienung der Anwenderseite

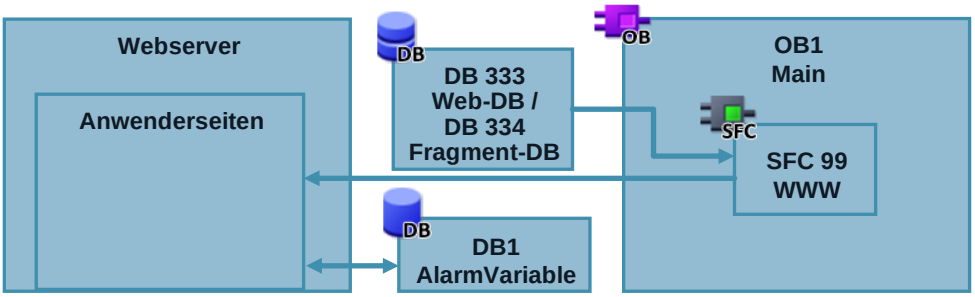
Schritt	Handlungsanweisung
1	Betätigen Sie eine der vier möglichen Schaltflächen, hier „alarm=0“.
2	Der Text „No Alarm! all okay.“ wird entsprechend der numerischen Zuordnung „alarm=0“ ausgegeben.

© Siemens AG 2019 All rights reserved

8.2.1 Aufbau des S7-Programms

In der nachfolgenden Abbildung ist die S7-Programmstruktur schematisch dargestellt. Der Wert der Variable „Alarm“ im DB1 wird durch die Feldauswahl der Anwenderseite gesetzt.

Abbildung 8-2: S7-Programm ENUM - Variablenwerte werden durch Text ersetzt



Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

Inhalt des AlarmVariable-DB (DB 1)

Nr.	Funktion
1	In diesem DB ist die Variable „alarm“ gespeichert, die von der Anwenderseite gelesen und beschrieben werden kann. Die Variable wird durch die Auswahl der Anwenderseite gesteuert (hier „alarm=0“) AlarmVariable Name | Data type | Start.. | Monitor value Static alarm | Int | 0 | 0

8.2.2 Aufbau der Web-Anwenderseite (HTML-Datei)

Nr.	Code-Abschnitt

1	Definition der ENUM-Variable in der Datei „_enumdefs.htm“
	`<!-- AWP_Enum_Def Name="MyAlarmEnum" Values='`
	`0:"No Alarm! all okay.",`
	`1:"Attention! Alarm 1 occurred!",`
	`2:"Attention! Alarm 2 occurred!",`
	`3:"Alarm 3!"' -->`
	`<!-- AWP_Enum_Def Name="abc" Values='0:"null", 1:"one"' -->`

2	Initialisierung der ENUM-Variable `<!-- AWP_In_Variable Name='"AlarmVariable".alarm' Enum="MyAlarmEnum" -->`

3	Der Wert der PLC-Variable wird gelesen und als Text angezeigt.
	`<p>The current value of variable "alarm" is:`
	`<strong> :="AlarmVariable".alarm:</strong></p>`

4	Beim Schreiben der PLC-Variable wird der zugewiesene Text verwendet. In die PLC-Variable wird der dazugehörige Wert (Ziffer) geschrieben.
	`<form method="post" action="">`
	`<p><input type="hidden" name='"AlarmVariable".alarm' value="No Alarm! all okay."></p>`
	`<p><input type="submit" value='"AlarmVariable".alarm=0'><p>`
	`</form>`

© Siemens AG 2019 All rights reserved

9 HTTP-Redirection nach Fehler (nur S7-1500)

9.1 Automatisierungsaufgabe

Wenn eine Variable einen bestimmten Wert überschritten hat, soll eine neue Anwenderseite mit entsprechender Meldung angezeigt werden.

Der Wert der Variable soll über ein Eingabefeld auf der Anwenderseite einstellbar sein.

Anforderungen an die Automatisierungsaufgabe

- Programmieren einer Anwenderseite mit einem Eingabefeld zum Einstellen eines Variablenwertes.
- Die ENUM-Variable wird vom STEP 7 Programm vorgegeben.
- Eine ENUM-Variable soll die URL (Uniform Resource Locator) steuern.

© Siemens AG 2019 All rights reserved

9.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Abbildung 9-1: Anwenderseite zur HTTP-Redirection nach Fehler

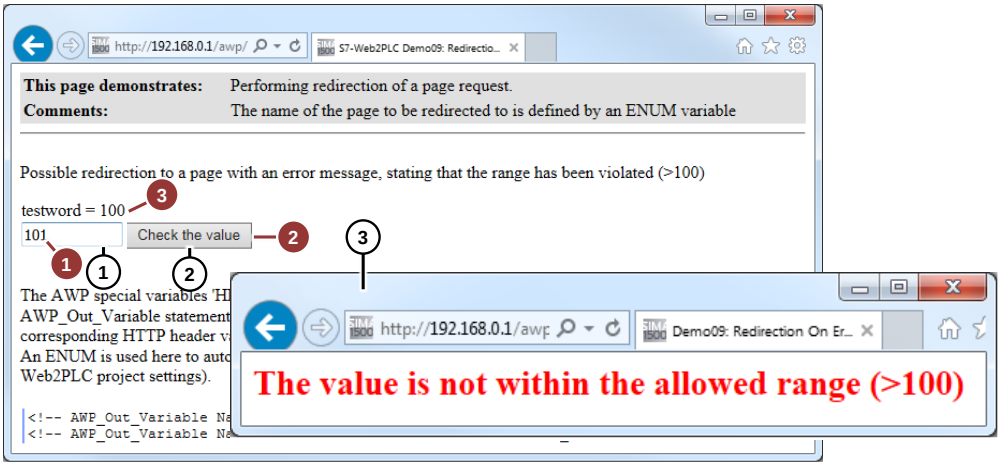


Tabelle 9-1: Anwenderseite zur HTTP-Redirection nach Fehler

Position	Beschreibung
1	In dieses Feld ist ein neuer Eingabewert einzutragen.
2	Mit der Schaltfläche „Check the value“ wird der Wert an die CPU gesendet.
3	Ist der Eingabewert größer als „100“, öffnet sich eine neue Anwenderseite. In diesem Beispiel ist die neue Anwenderseite mit rotem Text hinterlegt.

Tabelle 9-2: Handlungsanweisungen zur Bedienung der Anwenderseite

Schritt	Handlungsanweisung
1	Geben Sie in das Eingabefeld zunächst einen Wert kleiner „100“ ein.
2	Betätigen Sie die Schaltfläche und wiederholen Sie die Eingabe mit „101“.
3	Der neue Wert wird angezeigt und die neue Anwenderseite öffnet sich, wenn der Eingabewert > „100“ entspricht.

Hinweis

Wird die geprüfte Variable von der Steuerung beeinflusst, ist die Anwenderseite zyklisch zu aktualisieren. Ansonsten wird der http-Redirect nicht ausgeführt.

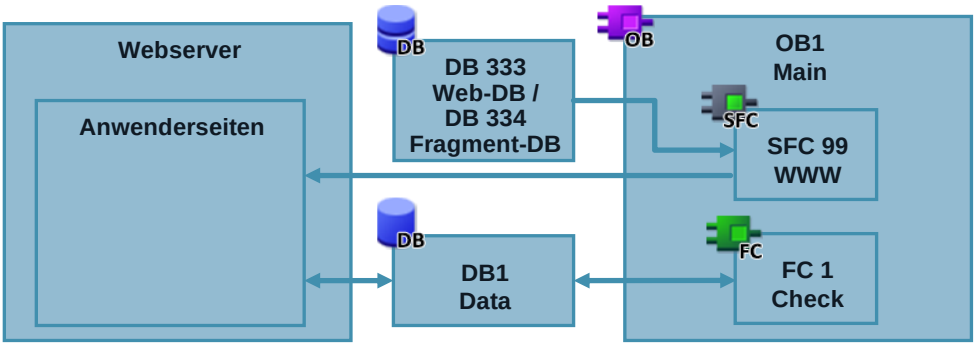
Um auf die Startseite zu gelangen, setzen Sie den Wert der Variable im TIA Portal unter „100“ und laden Sie die Anwenderseite neu.

© Siemens AG 2019 All rights reserved

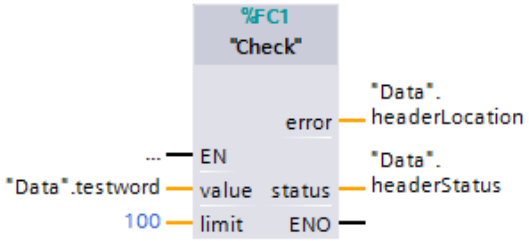
9.2.1 Aufbau des S7-Programms

In der nachfolgenden Abbildung ist die S7-Programmstruktur schematisch dargestellt. Über einen Werteabgleich im FC1 wird die Anwenderseite umgeschaltet.

Abbildung 9-2: S7-Programm zur HTTP-Redirection nach Fehler



Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.
2	Die Funktion „Check“ ermittelt, ob das Limit von „100“ überschritten wurde. Ist dies der Fall werden die Variablen „headerLocation“ auf „1“ und der „headerStatus“ auf „302“ gesetzt. Dadurch wird auf die Anwenderseite mit der Fehlermeldung umgeschaltet. Wurde der Wert nicht überschritten, werden beide Variablen auf „0“ gesetzt. 

Inhalt des Data-DB (DB1)

Nr.	Funktion
1	Im DB1 werden die Variablen für den FC 1 gespeichert. Ist der Wert der Variable „testword“>100, wird die Anwenderseite umgeschaltet. Data Name / Data type / Start value / Monitor value Static testword Int 0 100 redirEnum Int 0 0 headerLocation Int 0 0 headerStatus Int 0 0

© Siemens AG 2019 All rights reserved

9.2.2 Aufbau der Web-Anwenderseite (HTML-Datei)

Nr.	Code-Abschnitt

1	Um die Location zu ändern, muss den Nummern, die in der Variablen „headerLocation“ gespeichert sind, die jeweilige URL zugewiesen werden.
	
	Dazu muss eine ENUM-Variable definiert werden.
	`<!-- AWP_Enum_Def Name="redirEnum" Values='`
	`1:"Demo09_redir.html",`
	`0:"Demo09.html"' -->`

2	Initialisierung der CPU-Variablen und Zuweisung der ENUM-Variable
	`<!-- AWP_In_Variable Name='"Data".testword' -->`
	`<!-- AWP_Out_Variable Name="HEADER:Status"`
	`Use='"Data".headerStatus' -->`
	`<!-- AWP_Out_Variable Name="HEADER:Location"`
	`Use='"Data".headerLocation' Enum="redirEnum" -->`

3	Beschreibung der Variable, die geprüft werden soll.
	`<form method="POST" action="">`
	`<input type="text" name='"Data".testword' size="10">`
	`<input type="submit" value="Check the value" >`
	`</form>`

© Siemens AG 2019 All rights reserved

10 Sprachumschaltung auf Anwenderseiten

10.1 Automatisierungsaufgabe

Auf den Anwenderseiten soll eine Sprachumschaltung (Deutsch/Englisch), über entsprechende Flaggensymbole realisiert werden.

Die Sprachumschaltung sollen Sie individuell durch weitere Sprachen ergänzen.

Anforderungen an die Automatisierungsaufgabe

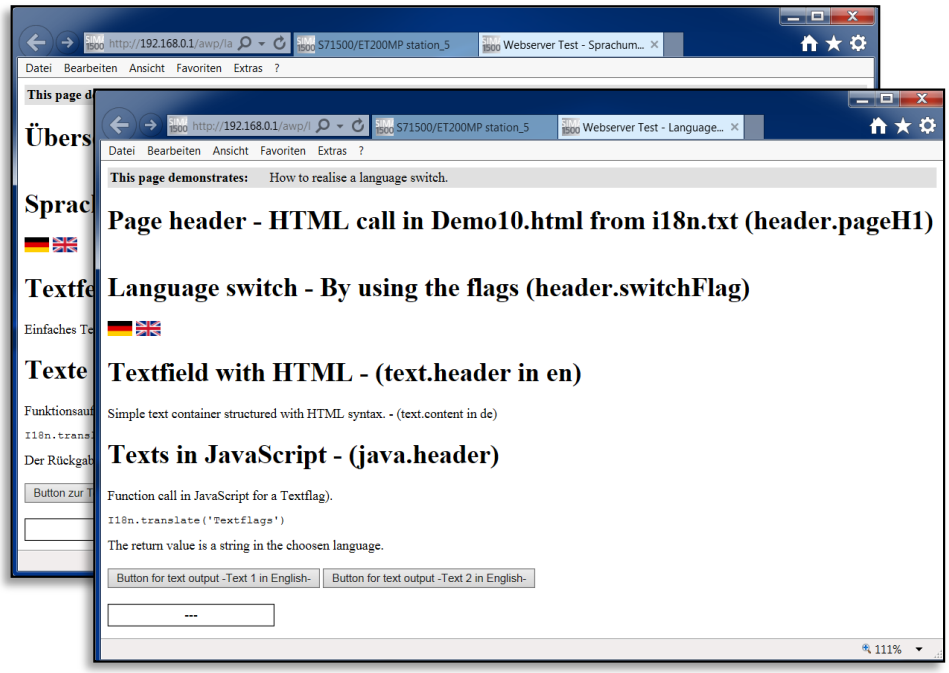
- Programmierung einer Anwenderseite ohne Texte.
- Erstellung einer Textdatei mit allen Texten in Deutsch und Englisch.
- Umschaltung der Sprache per Landesflagge.
- Übersetzung von Texten, die von JavaScript gewählt wurden.
- Synchronisieren der Sprachumschaltung der Anwenderseiten mit der Sprachumschaltung der Web-Standardseiten.
- Erweiterbarkeit der Sprachumschaltung.

10.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Auf der Anwenderseite schalten Sie zwischen den Sprachen Deutsch und Englisch um. Klicken Sie hierzu auf die jeweilige Landesflagge.

Abbildung 10-1: Anwenderseite zur Sprachumschaltung

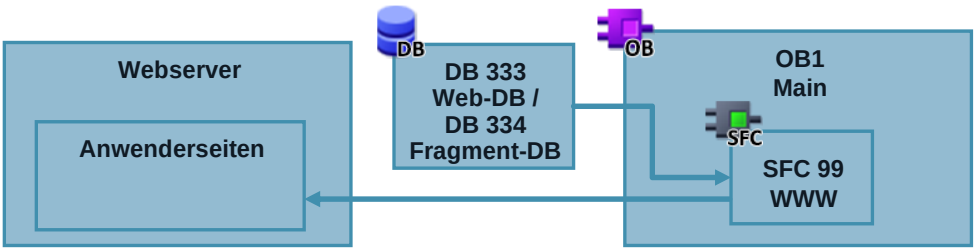


© Siemens AG 2019 All rights reserved

10.2.1 Aufbau des S7-Programms

Im S7-Programm wird ausschließlich die WWW-Funktion aufgerufen.

Abbildung 10-2: S7-Programm zur Sprachumschaltung



Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

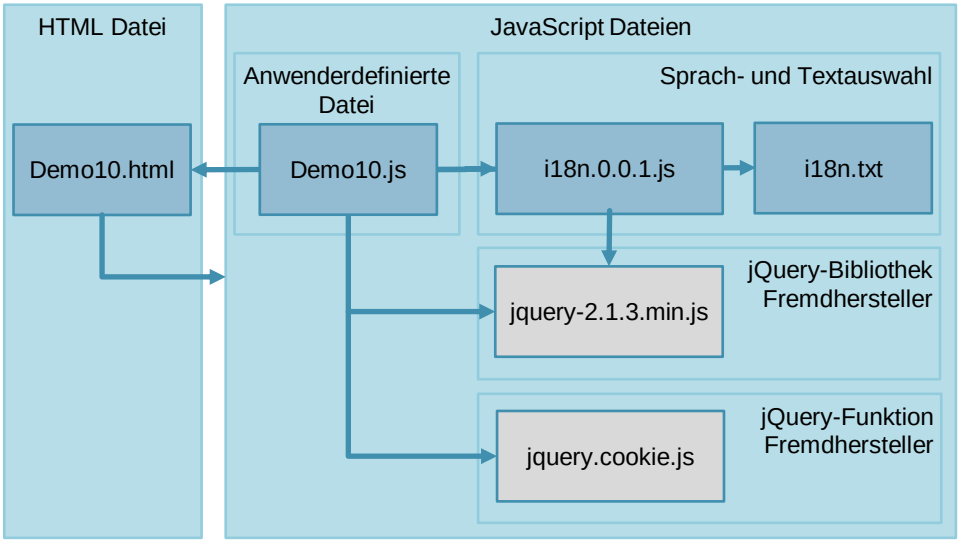
10.2.2 Struktur der Web-Anwenderseite

Die grundlegende Dateistruktur der Anwenderseiten zeigt die nachfolgende schematische Übersicht.

Die nachfolgend wiederkehrende Bezeichnung „i18n“ steht für einen Prozess in der Softwareentwicklung, um die Sprachumschaltung unabhängig vom eigentlichen Programmcode zu realisieren. Die Bezeichnung „i18n“ ergibt sich aus der Internationalisierung, wobei zwischen „i“ und „n“ 18 Buchstaben stehen.

Um den Code des Skriptes „Demo10.js“ effizient, und die Funktion der Anwenderseiten browserübergreifend kompatibel zu programmieren, ist die jQuery-Bibliothek für JavaScript eingebunden.

Abbildung 10-3: Schematische Übersicht der Web-Anwenderseite



© Siemens AG 2019 All rights reserved

Demo10.html

Das HTML-Dokument definiert den Aufbau der Anwenderseite und die Aufrufe der verwendeten JavaScript-Dateien, siehe Kapitel 10.2.2.1.
Die Sprachzuordnung wird in der Dateistruktur übergreifend durch die Identifier „de“ und „en“ gewährleistet.

Demo10.js

Enthält die interaktiven Funktionen, mit denen die Benutzereingaben umgesetzt werden siehe Kapitel 10.2.2.2.
Ebenso ist das Verhalten des verwendeten Cookie definiert. Der Codesyntax des JavaScripts ist durch die jQuery-Bibliothek unterstützt.

i18n.txt

Die Textdatei „i18n.txt“ enthält die mehrsprachigen Text-Elemente, die „Demo10.html“ und die Skriptfunktionen aus „i18n:0.0.1.js“ aufgerufen werden.

i18n.0.0.1.js

Enthält die Funktionen zur Sprachumschaltung und zum Transfer der sprachgebundenen Texte aus „i18n.txt“ in das HTML-Dokument.

Abbildung 10-4: Code-Element zur Sprachumschaltung

```
// replace text of all elements with data-i18n attribut
function translateLabels() {
    var item, text, key;
    $("[data-i18n]").each(function() {

// Load languagge definition file to var dictionary
function load() {
    var filename = "script/i18n.txt";
    //$.ajax({ type: "POST", url: url, data: "", dataType: "text"})
```

jquery.cookie.js

Das JavaScript bietet Funktionen zum Definieren und Verwalten von Cookies.

Das JavaScript „Demo10.js“ nutzt hierbei diese Funktionen und erstellt, oder verändert das bestehende Cookie der Web-Standardseite, siehe Kapitel 10.2.2.2.
Weiter Informationen zum hier verwendeten JavaScript entnehmen Sie der folgenden Webseite:

- https://github.com/carhartl/jquery-cookie
  - https://github.com/carhartl/jquery-cookie/blob/d72bb07e29962330db61aae1eda07f6312419840/src/jquery.cookie.js
  - https://github.com/js-cookie/js-cookie/tree/master/src

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei jQuery-Bibliothek, ermöglicht ein effizienteres Programmieren mit JavaScript. Weitere Informationen zur jQuery-Bibliothek finden Sie im Kapitel 4.3

© Siemens AG 2019 All rights reserved

10.2.2.1 Aufbau - Demo10.html

Nr.	Code-Abschnitt

1	Im <head> des HTML-Dokuments werden die genutzten JavaScripte mit der dargestellten Syntax eingebunden:
	
	`<head>`
	`<title data-i18n="title"></title>`
	`<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
	`<!--<script type="text/javascript" src="../file.js"></script>-->`
	`<script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>`
	`<script type="text/javascript" src="script/jquery.cookie.js"></script>`
	`<script type="text/javascript" src="script/i18n_0.0.1.js"></script>`
	`<script type="text/javascript" src="script/Demo10.js"></script>`
	`</head>`
	
	Die Eigenschaft „src“ bestimmt den jeweiligen Dateipfad.

2	Die Header der Webseite (Webseitenüberschrift) ist nicht in die Sprachumschaltung eingebunden. Sie besteht aus den beiden strukturierenden Klassen „demohead“ und „demorow“ und den beiden untenstehenden Textelementen mit entsprechender Formatierungsanweisung. **This page demonstrates:** How to realise a language switch.

	Die Texte stehen hierbei direkt im HTML-Code und werden nicht übersetzt!
	
	`<body>`
	`<!-- Head for the Webpage ----------`
	`<style>`
	`.demohead { background-color:#E0E0E0; width:100%}`
	`.demorow { }`
	`.democoltitle { width: 189px;font-weight:bold;vertical-align:top;}`
	`.democoltext { }`
	`</style>`
	
	`<table class="demohead" >`
	`<tr class="demorow" >`
	`<td class="democoltitle" >This page demonstrates:</td>`
	`<td class="democoltext" >How to realise a language switch.</td>`
	`</tr>`
	`</table>`

3	Nachfolgend ist die Sprachumschaltung über die Flaggensymbole dargestellt. Die Flaggensymbole sind mit der Klasse „language“ verbunden.

	`<span>`
	`<img src="Images/flag-deutschland.png"`
	`class="language" data-lang="de" data-i18n="lang.de"`
	`width="30px" height="18px" />`
	`</span>`
	Den Schaltelementen für den Sprachwechsel ist die Klasse „language“, mit der Eigenschaft (data-lang=“..“) zugewiesen, hier z. B. „Deutsch“.
	Die Eigenschaft (data-lang) bestimmt die Sprache, in die gewechselt werden soll.
	Hinweis: Für weitere Sprachen ist der Codeabschnitt gleich aufgebaut.

© Siemens AG 2019 All rights reserved

Nr.	Code-Abschnitt
4	Wie der Text in der Anwenderseite, oberhalb der Flaggensymbole, werden auch die nachfolgenden HTML-Textelemente zwischen „de“ und „en“ umgeschaltet. **Textfield with HTML - (text.header in en)** Simple text container structured with HTML syntax. - (text.content in de)

	Die Eigenschaft „data-i18n“ verweist auf einen eindeutig identifizierbaren Namen z. B. „text.header“ in der Datei „i18n.txt.
	
	`<div class="container" style="clear: left">`
	`  <h1> <span data-i18n="text.header"> </span></h1>`
	`  <span data-i18n="text.content"> </span>`
	`</div>`

	Verweis aus Demo10.html auf Datei „I18n.txt“
	In der Datei „i18n.txt“ werden den identifizierenden Namen wie („text.header“) die Texte der jeweiligen Sprache zugewiesen.
	Somit steht in der Anwenderseite „Textfield mit HTML…“ anstelle von „text.header“.
	
	`"text.header" : {`
	`  "de": "Textfeld mit HTML - (text.header in de)",`
	`  "en": "Textfield with HTML - (text.header in en)"`
	`},`

© Siemens AG 2019 All rights reserved

10.2.2.2 Aufbau - Demo10.js

Nr.	Code Abschnitt
1	Das Skript enthält zunächst die Variablendeklaration für die Funktionen der Schaltelemente der HTML-Seite. `var elements = {};`

2	Die Zuordnung der Texte zu den Buttons und in das Ausgabefeld, ist durch den folgenden Code-Abschnitt realisiert.
	
	`$.init = function(){`
	`  elements.language = $(".language");`
	`  elements.button1 = $("#textByJavaButton1");`
	`  elements.button2 = $("#textByJavaButton2");`
	`  elements.output = $("#textByJava");`
	
	Links: elements.button1
	Rechts: elements.button2
	Unten: elements.output
	
	Button for text output -Text 1 in English- Button for text output -Text 2 in English-
	--- 	

3	Wird auf ein Element mit (class="language") geklickt, wird der Eigenschaftswert (Sprache) aus „data-lang" an die Funktion „I18n.setLanguage" im Skript „i18n_0.0.1.js" übergeben.
	
	Beispiel:
	`elements.language.click(function () {`
	`  I18n.setLanguage( $(this).attr("data-lang") );`
	
	Gleichzeitig wird die Sprachauswahl im Cookie „siemens_automation_language" gespeichert und mit der Web-Standardseite synchronisiert.
	
	`$.cookie("siemens_automation_language", $(this).attr("data-lang"),`

	Beim ersten Aufruf der Anwenderseite wird der Wert der Funktion "I18n.setLanguage" aus dem Cookie übernommen.
	Wenn kein Cookie vorhanden ist, wird die Sprache per Default auf „en" gesetzt.
	
	`if($.cookie("siemens_automation_language") == null) {`
	`  $.cookie("siemens_automation_language", "en", { path : '/'})`
	`}`
	`I18n.setLanguage($.cookie("siemens_automation_language"));`

4	Bei einem Klick gibt die Funktion den Text entsprechend dem identifizierenden Namen in der aktuellen Sprache aus.
	Die Eigenschaft „data-i18n" wird auf die aktuelle ID „java.output1" gesetzt, damit der Text bei der nächsten Sprachumschaltung übersetzt wird.
	
	`elements.button1.click(function () {`
	`  elements.output.html( I18n.translate( "java.output1" ) );`
	`  elements.output.attr("data-i18n","java.output1");`
	`});`

5	Hinweis: Weitere Informationen zur Textänderung mit JavaScript sind im Code der Datei „Demo10.js" enthalten.

© Siemens AG 2019 All rights reserved

10.3 Erweiterung des Beispiels

Erweitern Sie das Beispiel um eine Sprache, indem Sie Code-Abschnitte duplizieren und anpassen. Erweitern Sie die Identifierer “de“, bzw. “en“ um die gewünschte Sprache, z. B. “fr“ für Französisch, oder „xx“ als Platzhalter einer beliebigen Sprache.

Neues Flaggensymbol hinzufügen

Für die neue Sprache benötigen Sie ein entsprechendes Flaggensymbol gemäß dem vorhandenen Beispiel, hier im PNG-Format. Kopieren Sie Ihr Flaggensymbol "xx.png" in den Ordner „…/Images/“

Demo10.html erweitern

1. Erweitern Sie den Code-Abschnitt entsprechend der Abbildung
2. Verweisen Sie im Pfad „Images“ auf das neue Flaggensymbol (xx.png)
3. Ändern Sie die Textzuordnung für „data-i18n“.

Abbildung 10-5: Erweiterung im HTML-Code

```
1
<span>
    <img src="Images/flag-unitedkingdom.png"
        class="language" data-lang="en" data-i18n="lang.en"
        width="30px" height="18px" />
</span>

<span>
    <img src="Images/xx.png"  2
        class="language" data-lang="xx" data-i18n="lang.xx"  3
        width="30px" height="18px" />
</span>
```

i18n.txt erweitern

Erweitern Sie die Textdatei „i18n.txt um weitere Sprachen. Die Textelemente für “de“ und “en“ sind den jeweiligen Namen (z. B. “lang.en“) zugeordnet. In der nachfolgenden Abbildung ist exemplarisch für diese Namen ein Code-Abschnitt dargestellt, den Sie für eine weitere Sprache duplizieren müssen. Erweitern Sie die Textelemente mit der gleichen Vorgehensweise.

1. Duplizieren Sie den Code-Abschnitt
2. Erweitern Sie den Code-Abschnitt mit dem neuen Spracheintrag.
3. Ändern Sie den Namen, zur Zuordnung des neuen Code-Abschnitts, hier „lang.xx“.

Abbildung 10-6: Erweiterungen in der Textdatei

```
1
"lang.en": {
    "de": "Englisch",
    "en": "English",
    "xx": "xxxxxxxxx)"  2
},

3
"lang.xx": {
    "de": "xdex",
    "en": "xenx",
    "xx": "xxxxxxxxx)"
},
```

© Siemens AG 2019 All rights reserved

11 Datenübertragung ohne wiederholten Seitenaufbau mit AJAX

Beim Datenaustausch zwischen einem Webserver und einem Webbrowser werden Webseiten vollständig neu übertragen.

Die Performance von schnell ablaufenden Aktualisierungen ist somit eingeschränkt.

Nachfolgend finden Sie eine Begriffsdefinition der beiden Verfahren zur Datenübertragung, die im Beispiel verwendet werden.

AJAX

AJAX „Asynchronous JavaScript and XML“ stellt ein Verfahren zur asynchronen Datenübertragung, zwischen einem Webbrowser und dem Webserver, bereit. Der Datenaustausch findet statt, ohne die Webseite selbst neu zu laden.

Inlineframe

Inlineframes werden zur Strukturierung von Webseiten verwendet, um Webinhalte von der umgebenden Webseite zu trennen.

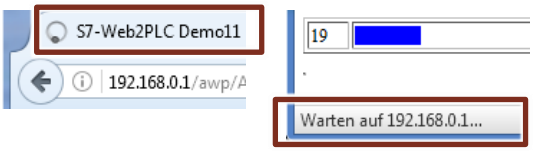
Die getrennten Inhalte stehen als eigenständige HTML-Dokumente zur Verfügung und werden von der Haupt-Webseite in einem definierten Bereich des Browsers mit definierter Größe angezeigt.

Die Inhalte des Inlineframes werden somit parallel zur umgebenden Seite aktualisiert.

Im Firefox-Browser wird der Unterschied zu AJAX deutlich, da im Tab ein Icon zum Laden der Webseite eingeblendet wird und im unteren Statusbereich die Meldung „Warten auf 192.168.0.1“.

Beim Verwenden von AJAX, werden diese Meldungen nicht dargestellt, da die Webseite selbst nicht neu geladen wird.

Abbildung 11-1: Webseitenaktualisierung im Firefox-Browser (Inlineframe)



© Siemens AG 2019 All rights reserved

11.1 Automatisierungsaufgabe

Es sollen Werte über zwei unterschiedliche Verfahren „AJAX“ und „Inlineframe“ eingelesen werden. Es sind hierzu zwei Anwenderseiten gleichen Aufbaus zu programmieren, die miteinander verlinkt sind.

Grundlegende Funktion:

- Einlesen des Eingabewertes für die Durchfluss-Geschwindigkeit „Velocity“.
- Anzeige des Füllstandes in einem Balken, abhängig von der Durchfluss-Geschwindigkeit.

Anforderungen an die Automatisierungsaufgabe

- Zyklisches Laden eines Wertes mit der Hilfe eines Inlineframes.
- Schreiben eines Wertes mit der Hilfe eines Formulars.
- Zyklisches Laden eines Wertes mit der Hilfe von AJAX.
- Schreiben eines Wertes mit der Hilfe von AJAX.

Voraussetzung zur Kommunikation zwischen STEP 7 und der Anwenderseite mit AJAX

Damit das Aktualisieren über AJAX funktioniert, binden Sie in den Einstellungen der CPU das Dateiformat mit dynamischem Inhalt ein.

Abbildung 11-2: Voraussetzung - Aktualisierung über AJAX mit „dat“ unter STEP 7

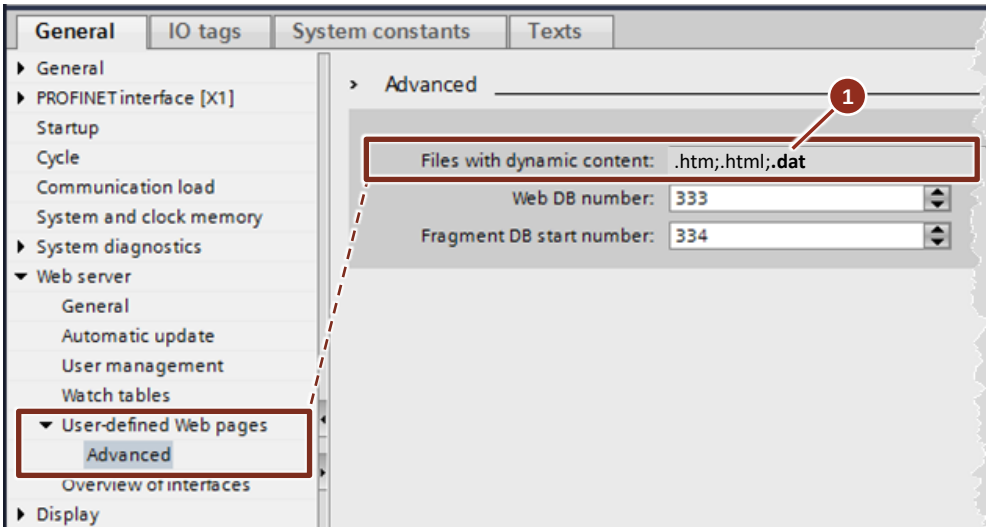


Tabelle 11-1: Handlungsanweisungen zur Erweiterung um ein Dateiformat

Schritt	Handlungsanweisung
1	Fügen Sie nach „.htm;.html;“ die Dateiendung „.dat“ in das Eingabefeld ein.

© Siemens AG 2019 All rights reserved

11.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Auf der Anwenderseite geben Sie einen Wert für die Durchfluss-Geschwindigkeit „Velocity“ ein. Der blaue Balken in der unteren Anzeige entspricht dem aktuellen Füllstand. Dieser wächst in Abhängigkeit zur Durchfluss-Geschwindigkeit.

Abbildung 11-3: Anwenderseite zum wiederholten Seitenaufbau mit AJAX / Inlineframe

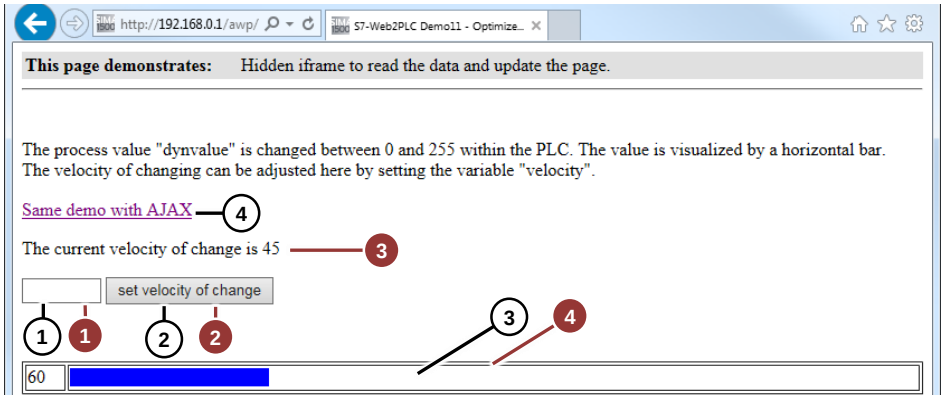


Tabelle 11-2: Anwenderseite zum wiederholten Seitenaufbau mit AJAX/Inlineframe

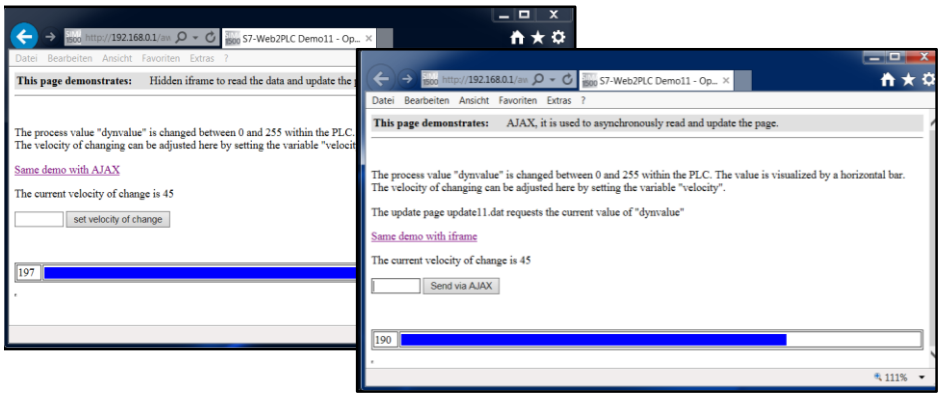
Position	Beschreibung
1	Eintrag neuer Werte für die Durchfluss-Geschwindigkeit „Velocity“
2	Die Schaltfläche überträgt den eingetragenen Wert an die Steuerung.
3	Abhängig vom Eingabewert wird der Balken in blau aufgefüllt.
4	Dieser Link schaltet die Anwenderseite zwischen Inlineframe und AJAX um.

Tabelle 11-3: Handlungsanweisungen zur Bedienung der Anwenderseite

Schritt	Handlungsanweisung
1	Geben Sie in das Eingabefeld einen neuen Wert ein.
2	Betätigen Sie die Schaltfläche und der Wert wird an die CPU übertragen.
3	Der neue Wert wird hier angezeigt.
4	Der Balken füllt sich nun mit der, vom Wert abhängigen Geschwindigkeit.

Es werden beiden Verfahren zur Übertragung der Werte gezeigt. Dabei wird auf die gleichen Variablen im S7-Programm zugegriffen, wodurch der Füllstand in beiden Anwenderseiten synchronisiert dargestellt wird.

Abbildung 11-4: Anwenderseiten Inlineframe (links), AJAX (rechts)

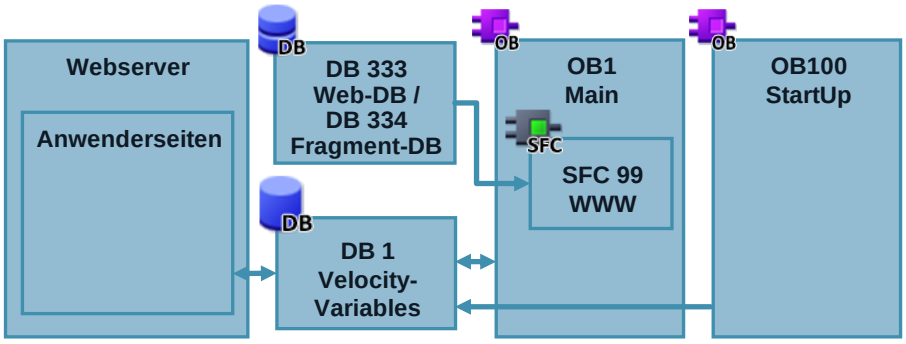


© Siemens AG 2019 All rights reserved

11.2.1 Aufbau des S7-Programms

Bei einem Neustart werden die Variablen im DB 1 auf die Startwerte zurückgesetzt. Im Umlauf des OB 1 wird die Anwenderseite durch den SFC 99 aktualisiert.

Abbildung 11-5: S7-Programm zum wiederholten Seitenaufbau mit AJAX / Inlineframe



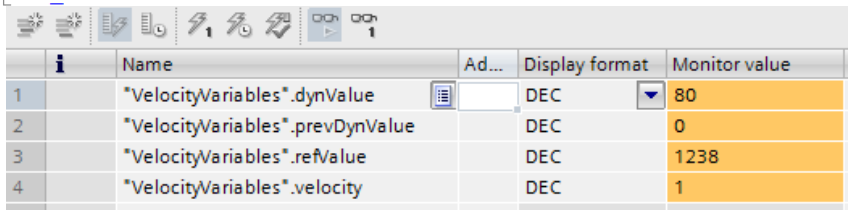
Funktionsweise des OB 100

Nr.	Funktion
1	Im OB100 werden die Variablen „DynValue“ und „Velocity“ zurückgesetzt.

	`"VelocityVariables".dynValue := 0;`
	`"VelocityVariables".velocity := 50;`
	• „dynValue“ bestimmt den Füllstand des Balkens.
	• „velocity“ bestimmt die Füllgeschwindigkeit des Balkens.

Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

2	Die Variable „refValue“ wird mit jedem OB 1 Zyklus um „1“ inkrementiert. Wird der Wert „2000“ erreicht, wird die Variable „dynValue“ um „1“ erhöht. Diese Variable vom Typ [Byte], skaliert die Füllstandanzeige auf 255 Schritte. Wird der Wert um 255+1 erhöht, wird die Variable „dynValue“ auf den Wert „0“ zurückgesetzt.
	`"VelocityVariables".refValue :=`
	`  "VelocityVariables".refValue + "VelocityVariables".velocity;`
	
	`IF "VelocityVariables".refValue >= 2000 THEN`
	
	`  "VelocityVariables".refValue := 0;`
	`  "VelocityVariables".dynValue := "VelocityVariables".dynValue + 1;`
	`END_IF;`
	 

	i	Name	Ad...	Display format	Monitor value
1		"VelocityVariables".dynValue		DEC	80
2		"VelocityVariables".prevDynValue		DEC	0
3		"VelocityVariables".refValue		DEC	1238
4		"VelocityVariables".velocity		DEC	1

Inhalt des VelocityVariables-DB (DB 1)

DB1 enthält die Variablen für dieses Beispiel.

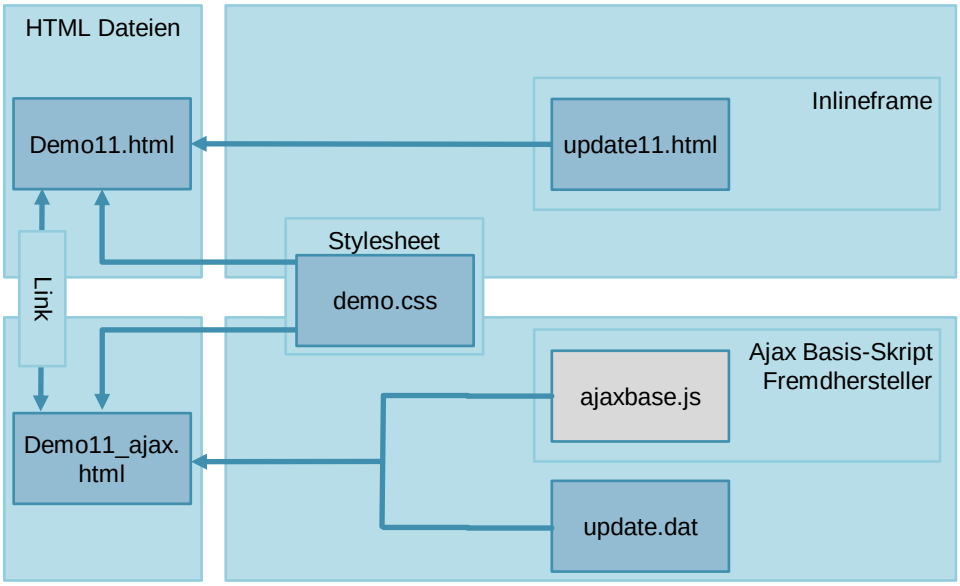
© Siemens AG 2019 All rights reserved

11.2.2 Struktur der Web-Anwenderseite

Die grundlegende Dateistruktur der Anwenderseiten zeigt die nachfolgende schematische Übersicht.

Diese Beschreibung legt den Fokus auf die Code-Elemente des JavaScripts, wodurch grundlegende HTML-Kenntnisse für dieses Anwendungsbeispiel vorausgesetzt sind.

Abbildung 11-6: Schematische Übersicht der Anwenderseite



Demo11.html

Definiert den Aufbau der Anwenderseite und beinhaltet die JavaScript Funktionen zur Aktualisierung der S7-Variablen über das Inlineframe. Die JavaScript-Funktionen stehen im <Head> der Anwenderseite.

update11.html (Inlineframe für Demo11.html)

Das Inlineframe ruft die Funktion „ForceUpdate“ aus der Datei „demo11.html“ auf, um die PLC-Variable „dynValue“ (Füllstand) kontinuierlich zu aktualisieren.

Demo11_ajax.html

Definiert den Aufbau der Anwenderseite und beinhaltet die JavaScript Funktionen für AJAX innerhalb des Anwenderseiten-<Head>.

Über die JavaScript Funktionen werden die PLC-Variablen aktualisiert.

update11.dat

Beinhaltet die Referenzvariablen „dynValue“ und „velocity“, die aus dem JavaScript unter Demo11_ajax.html heraus aufgerufen werden. (Funktion „DoHttpRequest“)

demo.css

Definiert das Stylesheet zu den Anwenderseiten für AJAX und Inlineframe.

ajaxbase.js

Das JavaScript beinhaltet die Funktionen zur Kommunikation und dem Datenaustausch zwischen verschiedenen Webbrowsern und Webservern.

© Siemens AG 2019 All rights reserved

11.2.2.1 Aufbau - Demo11.html (Inlineframe)

Nr.	Demo.html
1	Nach jeder Seitenaktualisierung wird die Funktion „Start“ ausgeführt. `<body onload="Start()">`

2	In der Funktion „Start“ werden zwei Funktionen aufgerufen.
	• „ForceUpdate“
	• „OnTimer“ Aufruf wird um 1000 Millisekunden verzögert.
	`function Start()`
	`{`
	`  ForceUpdate(:="VelocityVariables".dynValue:);`
	`  setTimeout("OnTimer()",1000);`
	`}`

	„ForceUpdate“ ermittelt die Länge des Balkens aus dem HTML-Dokument und überträgt diese.
	Für die Balkenanzeige werden Formatierungen für FireFox und IE durchgeführt.
	`function ForceUpdate(val)`
	`{`
	`  var width, barval;`
	`  var tabelem;`
	`  tabelem = parent.document.getElementById("bar");`
	`  width = tabelem.parentNode.clientWidth;`
	`  barval = ((val*width)/256);`
	`  if (barval == 0) barval = 1;`
	`  tabelem.style.width = Math.floor(barval)+"px";`
	`  var td = parent.document.getElementById("td1");`
	`  if (td.textContent)`
	`  {`
	`  td.textContent = val+"";`
	`  }`
	`  else`
	`  {`
	`  td.innerHTML = val+"";`
	`  }`
	`  g_bPageRequested = false;`
	`}`

	„OnTimer“ aktualisiert den Wert von „DynValue“ alle 200ms.
	Hierzu ruft ein Inlineframe die Funktion „ForceUpdate“ auf.
	`function OnTimer()`
	`{`
	`  if (! g_bPageRequested)`
	`  {`
	`  g_bPageRequested = true;`
	`  window.frames["hiddenFrame"].document.location.replace('update11.html');`
	`  }`
	`  setTimeout("OnTimer()", 200);`
	`}`

3	Zum Schreiben des Wertes „Velocity“ wird ein Formular verwendet, das Werte über eine „submit“-Schaltfläche sendet.

© Siemens AG 2019 All rights reserved

11.2.2.2 Aufbau - Demo11_ajax.html (AJAX)

Nr.	Demo11_ajax.html
1	Nach jeder Seitenaktualisierung wird die Funktion „Start“ ausgeführt. `<body onload="Start()">`

2	In der Funktion „Start“ werden drei Funktionen aufgerufen.
	• „DetermineBrowser“ ermittelt den Browsertyp (Mozilla, IE11, Chrome, etc.). Die Funktion ist in der Datei „ajaxbase.js“ enthalten.
	„ForceUpdate“ gleiche Funktion, wie unter Verwendung des Inlineframe zum Ermitteln der Balkenlänge.
	(Siehe Kapitel 11.2.2.1)
	• „OnTimer“ Aufruf wird um 1000 Millisekunden verzögert.
	
	`function Start()`
	`{`
	`  DetermineBrowser();`
	`  ForceUpdate(:="VelocityVariables".dynValue:);`
	`  setTimeout("OnTimer()",1000);`
	`}`

	„OnTimer“ aktualisiert den Wert von DynValue alle 200ms über den Funktionsaufruf „DoHttpRequest“ aus der Datei: „ajaxbase.js“.
	Der Funktion werden dabei vier Werte übergeben.
	1. Das Objekt, das die aktuelle Funktion aufgerufen hat.
	2. Die URL „update11.dat“ mit den zu aktualisierenden Variablen.
	3. Die Funktion „UpdateCallback“, die den Wert und den Statuscode weiterverarbeitet.
	4. Der Wert „true“ gibt an, dass es sich um eine asynchrone Datenübertragung handelt.
	
	Sind alle Werte aktualisiert, wird die Funktion „UpdateCallback“ aufgerufen.
	`function OnTimer()`
	`{`
	`  if (! g_bPageRequested)`
	`  {`
	`  g_bPageRequested = true;`
	
	`  DoHttpRequest(this, "update11.dat",  UpdateCallback, true);`
	`  }`
	`  setTimeout("OnTimer()", 200);`
	`}`

© Siemens AG 2019 All rights reserved

Nr.	Demo11_ajax.html

4	Die Funktion „UpdateCallback“ verarbeitet die gelesenen Werte und den Statuscode.
	Die Variable „response“ enthält die aktualisierten Werte der PLC-Variablen. Zur Verarbeitung in einem JavaScript müssen die Werte geteilt und neu zugeordnet werden.
	In der Variable „status“ ist der aktuelle HTTP-Statuscode enthalten.
	Ist der HTTP-Statuscode kleiner 300 wird die Funktion „ForceUpdate“ aufgerufen und die Länge des aktualisierten Balkens berechnet.
	
	`function UpdateCallback(obj, response, status) {`
	`  var ok;`
	`  var results = response.split(" ");`
	`  var signs = results[0].split("");`
	`  var i;`
	`  var count = 0;`
	`  for (i = 0; i < signs.length; i++) {`
	`  if (true == isNaN(signs[i])) {`
	`  count = count + 1;`
	`  }`
	`  else {break;}`
	`  }`
	`  dynValue = results[0].substr(count, signs.length);`
	
	`  var dynValueInt = parseInt(dynValue);`
	
	`  if (status < 300) {`
	`  document.getElementById('veloDiv').innerHTML = results[1],`
	`  ForceUpdate(dynValueInt);`
	`  g_bPageRequested = false;`
	`  setTimeout("OnTimer()", 200);`
	`  return;`
	`  }`
	`  if (status == 503) {`
	`  ok = confirm(dynValueInt);`
	`  } else {`
	`  ok = confirm("FAILED: HTTP error " + status);`
	`  }`
	`  g_bPageRequested = false;`
	`  if (ok) {`
	`  setTimeout("OnTimer()", 1000);`
	`  }`
	`}`

© Siemens AG 2019 All rights reserved

Nr.	Demo11_ajax.html
5	Durch das Betätigen der „Send via AJAX“-Schaltfläche wird die Funktion „send_ajax_request“ aufgerufen. The current velocity of change is [ ] Send via AJAX `<input type="button" onclick="send_ajax_request('%22Velocity%22', 'velocityField')" value="Send via AJAX">`
	Mit der Funktion „send_ajax_request“ wird der eingegebene Wert an die CPU übertragen.

```
function send_ajax_request(variable, fieldId)
{
    if (window.XMLHttpRequest)
    {
        req = new XMLHttpRequest();
    }
    else if (window.ActiveXObject)
    {
        req = new ActiveXObject("Microsoft.XMLHTTP");
    }
    else
    {
        alert("Der Browser unterstuezt kein Ajax");
    }
    var value = document.getElementById(fieldId).value;

    var req_url = "?"+variable+"="+value+"&"+Math.random();
    //debug alert(req_url);
    req.open("GET", req_url, false);
    req.onreadystatechange = ajax_callback;
    req.send(null);
}
```

© Siemens AG 2019 All rights reserved

12 Graph aus Daten eines Datalogs anzeigen

12.1 Automatisierungsaufgabe

Mit der S7-Steuerung sollen Werte von Variablen in einem Datalog gespeichert werden.

Datalogs sind CSV-Dateien, die auf der Memory Card, bzw. den internen Ladespeicher der CPU geschrieben werden.

Die Datenmenge ist hierbei abhängig vom freien Speicher.

Die Daten aus dem Datalog sollen in einer Anwenderseite als Graph angezeigt werden. Die Daten werden erst aktualisiert, wenn eine Schaltfläche [Load new data] angeklickt wird.

Hinweis

In diesem Beispiel werden Sinus- und Cosinus- Werte zyklisch auf die SMC Karte, bzw. in den Speicher der CPU geschrieben.

Bezüglich der Schreib-/Lesezyklen beachten Sie den Beitrag:

https://support.industry.siemens.com/cs/ww/de/view/109482591

Alternativ kann ein Ringpuffer für die Werte programmiert werden, mit dem eine begrenzte Anzahl von Werten gespeichert und im Graphen angezeigt wird.

© Siemens AG 2019 All rights reserved

Anforderungen an die Automatisierungsaufgabe

- Zyklisches Auslesen eines Datalogs.
- Chronologische Sortierung der ausgelesenen Daten.
- Anzeige der Daten als Graph.

12.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Auf der Anwenderseite werden die Daten aus dem Datalog, in einem Linien-Graph angezeigt. Die CSV-Datei des Datalog heißt „SinusUndCosinus.csv“.

Abbildung 12-1: Anwenderseite zur Datenanzeige aus Datalog in einem Linien-Graph

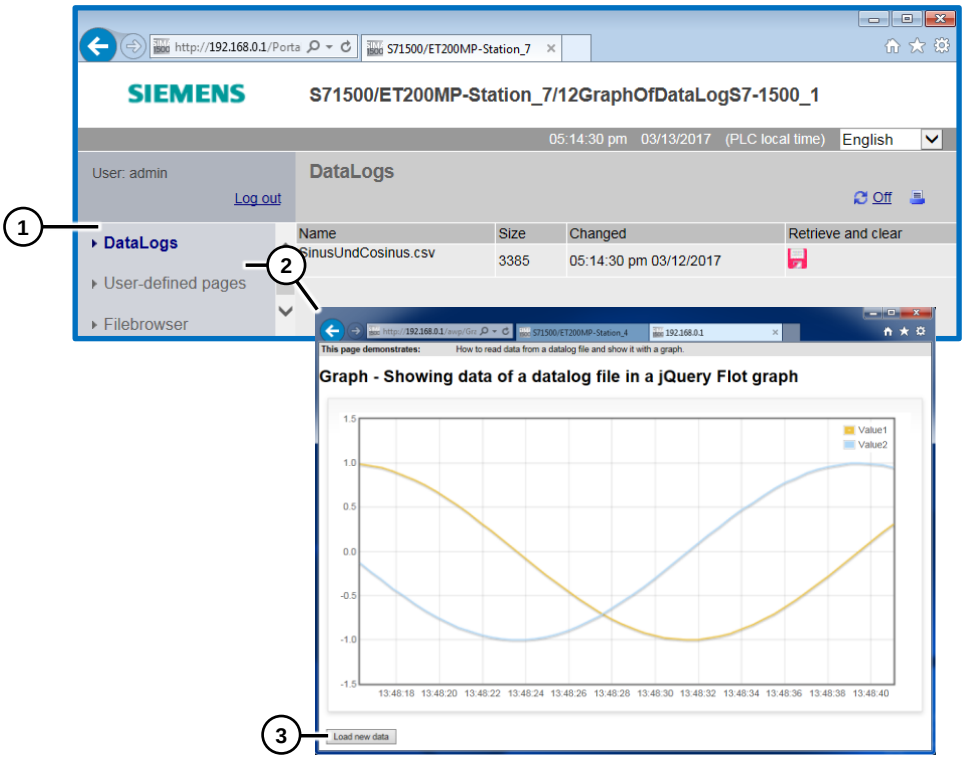


Tabelle 12-1: Anwenderseite zur Datenanzeige aus Datalog in einem Linien-Graph

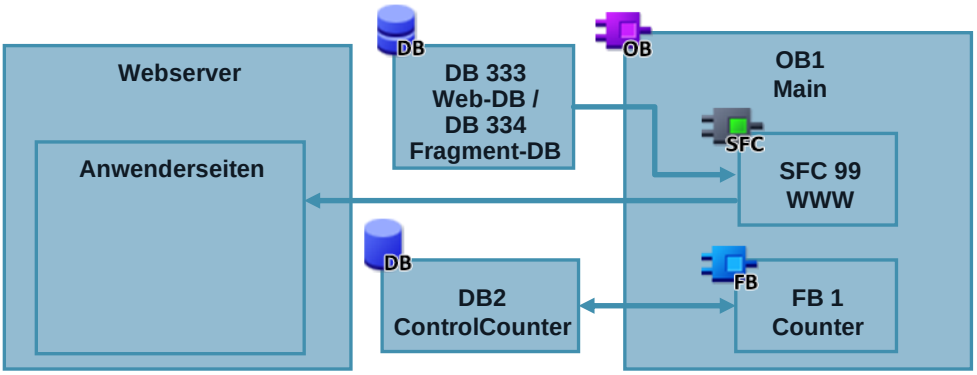
Position	Beschreibung
1	Datalog als CSV Datei „SinusUndCosinus.csv“ auf MemoryCard abgelegt
2	Angezeigter Graph in Anwenderseite (Quelldatei ist „SinusUndCosinus.csv“)
3	Schaltfläche zur Aktualisierung der Daten [Load new data]

© Siemens AG 2019 All rights reserved

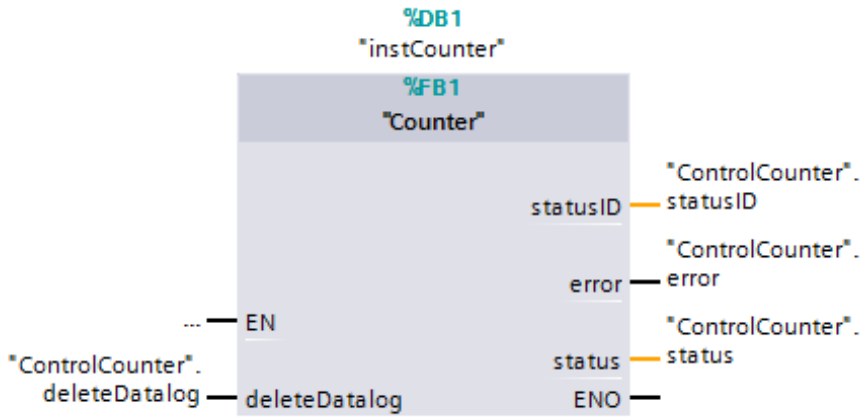
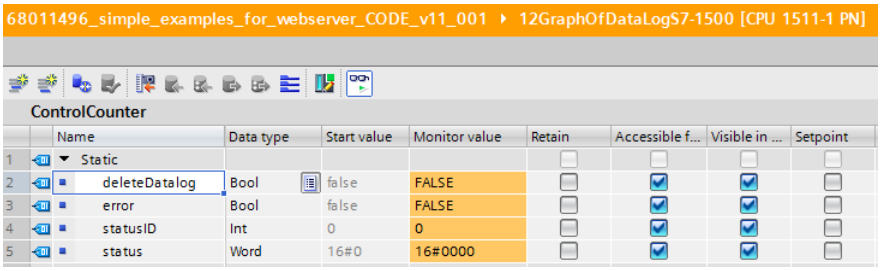
12.2.1 Aufbau des S7-Programms

Im Umlauf des OB 1 wird die Anwenderseite durch den SFC 99 aktualisiert.

Abbildung 12-2: S7-Programm zur Datenanzeige aus Datalog in einem Linien-Graph



Funktionsweise des OB 1

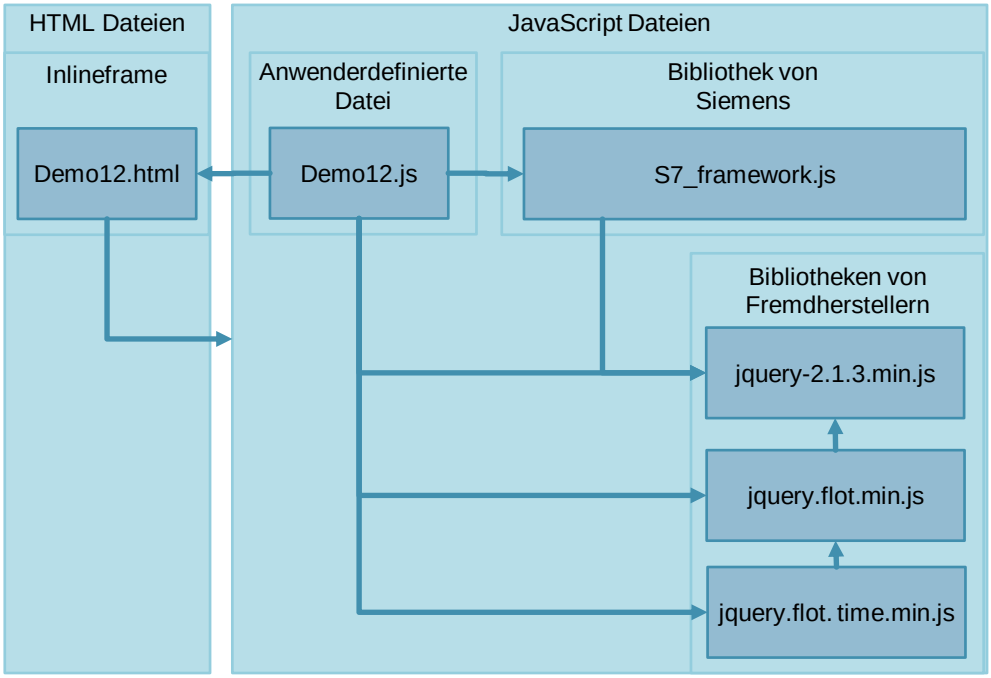
Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.
2	Im FB 1 werden Sinus- und Cosinus-Werte berechnet und ein Datalog wird erstellt. Parameter und Name der Datalog-CSV Datei werden in statische Variablen geschrieben. Die Sinus- und Cosinus-Werte werden in diesem Datalog zyklisch gespeichert.  Im DB 2 wird der Status und Errorcode abgelegt und der Eingangsparameter zum Löschen des Datalog Inhalts.  68011496_simple_examples_for_webserver_CODE_v11_001 ▸ 12GraphOfDataLogS7-1500 [CPU 1511-1 PN] ControlCounter Name / Data type / Start value / Monitor value / Retain / Accessible f... / Visible in ... / Setpoint 1 Static 2 deleteDatalog Bool false FALSE 3 error Bool false FALSE 4 statusID Int 0 0 5 status Word 16#0 16#0000

© Siemens AG 2019 All rights reserved

12.2.2 Struktur der Web-Anwenderseite

Die grundlegende Dateistruktur des Beispiels zeigt die nachfolgende schematische Übersicht.

Abbildung 12-3: Schematische Übersicht der Anwenderseite



© Siemens AG 2019 All rights reserved

Demo12.html mit (Graph – Container)

Definiert den Aufbau der Anwenderseite und beinhaltet einen Container für die Anzeige des Graphen. Im HTML-Dokument werden die JavaScript-Dateien eingebunden, um den Graphen anzuzeigen.

Abbildung 12-4: Demo12.html mit Graph

Tabelle 12-2: Demo12.html mit Graph

Position	Beschreibung
1	Demo12.html
2	Container des Graph mit Zugriff auf die Daten des Datalog

Demo.js

In der Datei „Demo12.js“ werden Funktionen aus den verschiedenen Bibliotheken aufgerufen um einen Graphen aus den Daten eines Datalogs (CSV-Datei) zu erstellen. Informationen zur Verarbeitung der Daten finden Sie im Quellcode.

jquery.flot.min.js und jquery.flot.time.min.js (Graph-Template)

Diese JavaScripte berechnen den Graphen in der Anzeige des Inlineframes.
In der Programmierung wurden Elemente aus der jQuery-Bibliothek verwendet.

Weitere Informationen finden Sie unter: http://www.flotcharts.org/

S7_framework.js

Das JavaScript beinhaltet eine Reihe von Funktionen zur Konvertierung und Verarbeitung von Daten aus einer SIMATIC S7.
Weitere Informationen zum S7-Framework finden Sie im Kapitel 4.3

S7_framework.css

Definiert das Stylesheet zu den HTML-Anwenderseiten

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei jQuery-Bibliothek, ermöglicht ein effizienteres Programmieren mit JavaScript. Weitere Informationen zur jQuery-Bibliothek finden Sie im Kapitel 4.3

© Siemens AG 2019 All rights reserved

12.2.2.1 Aufbau – Demo12.html

Nr.	Demo12.html Code-Abschnitt

1	Im <head> des HTML-Dokuments werden die genutzten JavaScripte mit der dargestellten Syntax eingebunden.
	
	`<head>`
	`<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
	`<link rel="stylesheet" type="text/css" href="CSS/S7_framework.css">`
	`<script type="text/javascript" src="script/jquery-2.1.3.min.js"></scr`
	`<script type="text/javascript" src="script/jquery.flot.min.js"></scri`
	`<script type="text/javascript" src="script/jquery.flot.time.min.js"><`
	`<script type="text/javascript" src="script/S7_framework_0.1.7.js"></s`
	`<script type="text/javascript" src="script/Demo12.js"></script>`
	`<script type="text/javascript"></script>`
	`</head>`
	
	Die Eigenschaft „src“ bestimmt den jeweiligen Dateipfad. Anwenderseiten können nicht direkt auf ein Datalog zugreifen.

2	Die Datalog-CSV Dateien sind auf der Web-Standardseite hinterlegt, die für den Zugriff aufgerufen werden muss.
	
	Der Graph wird in einem Container Element aufgerufen.
	
	`<h1><span>Graph - Showing data of a datalog file in a jQuery Flot graph</span></h1>`
	`<div class="graph-container">`
	`  <div id="graph-placeholder"></div>`
	`</div>`
	
	Das "div"-Tag "graph-container" ist ein Platzhalter für den Graphen.
	Das JavaScript „Demo12.js“ setzt den Graphen im Platzhalter zusammen.

© Siemens AG 2019 All rights reserved

12.2.2.2 Aufbau – Demo12.js

Nr.	Code-Abschnitt

1	**Ermitteln des CPU-Typs (S7-1200 oder S7-1500)**
	Im JavaScript „Demo12.js“ wird in „$(document).ready(function(){}“ eine Funktion ausgeführt um zu ermitteln, ob eine S7-1200 oder S7-1500 verwendet wird.
	`$(document).ready(function(){`
	`  if (plcType != "1200" && plcType != "1500")`
	`  {`
	`  $.ajax({type:"GET",url:"../../../Portal/Intro.mwsl",data:"",dataType:"text"})`
	`  .done(function(webpageData){`
	`  var search12 = webpageData.search("CPU12");`
	`  var search15 = webpageData.search("CPU15");`
	`  if (search12 >= 0)`
	`  {`
	`  plcType = "1200";`
	`  $.init();`
	`  }`
	`  else if (search15 >= 0)`
	`  {`
	`  plcType = "1500";`
	`  $.init();`
	
	Alternativ haben Sie die Möglichkeit in „$(document).ready(function[]{}“ direkt die Funktionen von „$.init“ aufzurufen.
	Ersetzten Sie „plcType“ durch „1200“ oder „1500“ inklusive Anführungszeichen

	Dem S7-Framework wird der Typ der PLC mit folgender Funktion übermittelt.
	`$.init = function(){`
	`  S7Framework.initialize(plcType, "");`
	`  S7Framework.readDataLog("SinusUndCosinus","Read Datalog failed", decodeCSV);`
	`}`
	
	Alternativ schreiben Sie für plcType „1200“, oder „1500“.
	`Beispiel:  S7Framework.initalize(“1200”, “”);`

2	**Datalog (Werte auslesen)** Die Funktion „readDataLog“ liest Datalogs aus. `S7Framework.readDataLog("SinusUndCosinus","Read Datalog Failed",decodeCSV);` Der Funktion ist der Name des Datalogs „SinusUndCosinus“ zu übergeben. Ein Text als Fehlermeldung ist optional. Der Funktionsname von „decodeCSV“ stellt das Ziel für die gelesenen Werte dar.
3	**Datalog (Werte verarbeiten)** In der Funktion „decodeCSV“, werden die Werte aus dem Datalog verarbeitet. `function decodeCSV(CSVdata){` Alle Zeitangaben werden in das Format [Date.UTC] umgewandelt und chronologisch geordnet. Die Daten werden für das Graph-Template aufbereitet und im JavaScript (jquery.flot.time.min.js) ausgewertet. In dieser Funktion werden auch die Eigenschaften des Graphen definiert. (Bsp. Punkte verbinden, keine Stufen im Graph)

© Siemens AG 2019 All rights reserved

13 Anzeigeelemente mit serieller Vektorgrafik

13.1 Programmieraufgabe

Es soll eine Auswahl von Anzeigeelementen mit serieller Vektorgrafik (SVG) erstellt werden. Diese Technik wird durch HTML5-fähige Browser unterstützt.

Anforderungen an die Automatisierungsaufgabe

- Erstellen von SVG-Elementen
- Dynamisieren von SVG-Elementen mit JavaScript

13.2 Funktionsmechanismen und Bedienung

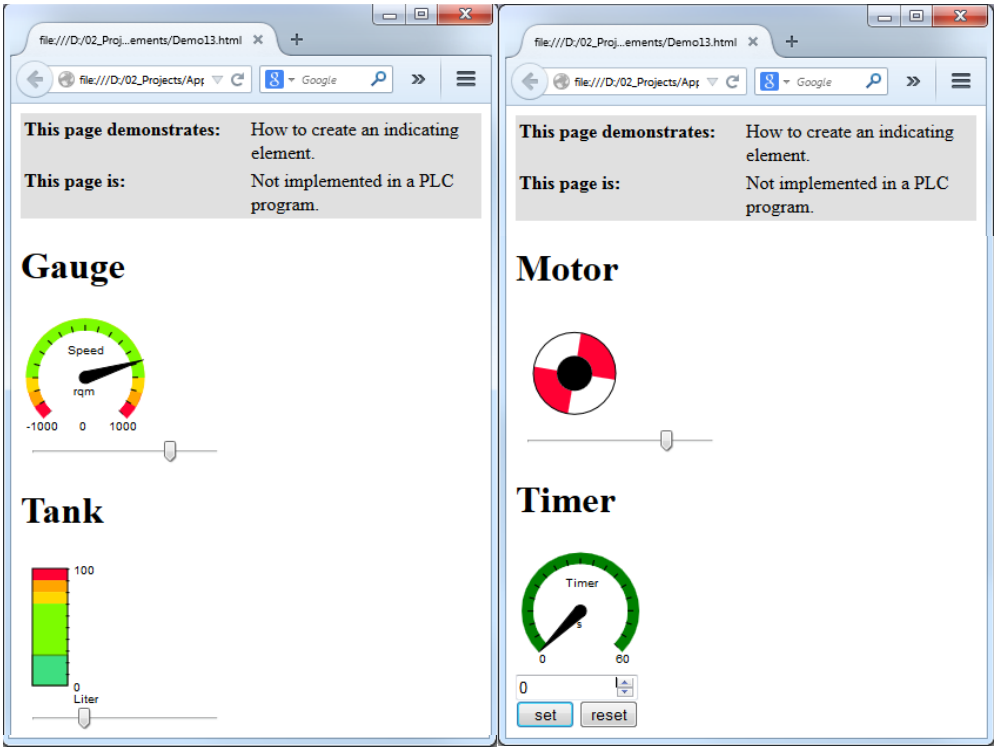
Hinweis

Dieses Beispiel enthält kein spezifisches S7-Programm. Die Anzeigeelemente greifen nicht explizit auf Daten einer SIMATIC S7 zu.
Die HTML-Datei finden Sie in dem Projektordner unter „html_indicatingElements“

Aufbau der Anwenderseite

Auf der Anwenderseite werden verschiedene Anzeigeelemente dargestellt, die mit einem Slider unterhalb der Anzeigeelemente verändert und gesteuert werden. Zur Anbindung an eine S7 ist der Slider durch die entsprechenden AWP-Variablen zu ersetzen.

Abbildung 13-1: Anwenderseite mit SVG-Anzeigeelementen serieller Vektorgrafik



© Siemens AG 2019 All rights reserved

13.2.1 Beispiel SVG-Element – Drehbewegung eines Motors

Programmieren von SVG-Elementen

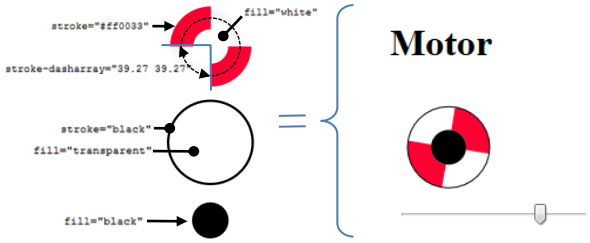
Die Form der SVG-Anzeigeelemente definieren Sie durch den HTML-Code. Für die Bewegungsfunktionen programmieren Sie ein JavaScript. Allgemein ändern Sie über Eingabevariablen die Position, oder Rotation von Kreis-, oder Rechteck-Elementen.

Die Eingabevariablen sind durch die Slider, unterhalb des jeweiligen SVG-Elements, realisiert.

Die folgende Abbildung zeigt ein exemplarisches Beispiel, für die Drehbewegung eines Motors und den dazugehörigen Code-Abschnitt im HTML-Dokument. Insgesamt setzt sich der Motor aus drei Kreiselementen zusammen.
Die Drehbewegung wird durch eine gestrichelte Linie (weiß, rot, weiß, rot) realisiert, die innerhalb eines zweiten Kreises für die Außengrenze rotiert. Der innere Kreis stellt die Nocke des Rotors dar.

Sie haben die Möglichkeit den Motor zu skalieren, indem Sie die Werte innerhalb der Anführungszeichen variieren. Hierbei müssen Sie stets den Wert für die gestrichelte Linie so anpassen, dass der Umfang des Kreises genau 4 Strichen entspricht. Der spezifische Parameter hierzu heißt „stroke-dasharray“.

Abbildung 13-2: Beispiel SVG-Anzeigeelement – Aufbau des Motors




```
<h1>Motor</h1>
<svg height="100" width="100">
    <circle id="motor.red"  cx="50" cy="50" r="25" stroke="#ff0033" stroke-dasharray="39.27 39.27"  stroke-width="20"   fill="white"
    <circle                 cx="50" cy="50" r="35" stroke="black"                                   stroke-width="2"    fill="transparent"
    <circle                 cx="50" cy="50" r="15" fill="black"                                     stroke-width="0"
</svg>
<br>
<input type="range" min="0" max="360" onclick="changeMotor(this.value)">
<script>
    function changeMotor(newValue)
    {
        document.getElementById("motor.red").setAttribute("transform", "rotate(" + newValue + ",50,50)")
    }
</script>
```

Tabelle 13-1: Anzeigeelemente zum Aufbau des Motors

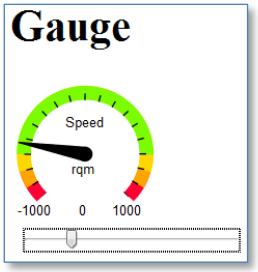
Position	Beschreibung
1	Kreise: Definition von Position, Größe und Farben für drei Kreiselemente.
2	Slider: Grafisches Element zur Eingabe von Werten.
3	Funktion: Zur Rotation des Motors

Hinweis

Zur Erweiterung Ihrer Applikation, kopieren Sie den Code-Abschnitt des SVG-Elements direkt in Ihr HTML-Dokument. Ersetzen Sie ggf. die Eingangsvariable des Slider-Elements entsprechend Ihrer Applikation.

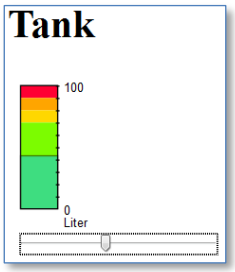
© Siemens AG 2019 All rights reserved

13.2.1.1 Aufbau – Demo13.html



Nr.	Code-Abschnitte aus (Gauge „Tacho“)
1	**<svg width=”110” height=”110”>** **…</svg>** Der Tag "svg" definiert den Grafikbereich. Der Ursprung des Grafikbereichs befindet sich links oben und die Ausdehnung der Koordinatenachsen entspricht der Grafikgröße in Pixeln. Im "svg"-Tag können folgende Elemente erstellt werden:
2	**<circle cx=”55” cy=”55” r=”40” />** Der Tag „circle“ erstellt einen Kreis. Die Eigenschaft „cx“ bestimmt die Position des Mittelpunktes auf der X-Achse, „cy“ auf der Y-Achse. Die Eigenschaft „r“ bestimmt den Radius.
3	**stroke=”#ff0033”** Die Eigenschaft „stroke“ bestimmt die Rahmenfarbe bzw. die Linienfarbe von SVG-Elementen.
4	**stroke-width=”20”** Die Eigenschaft „stroke-width“ bestimmt die Rahmendicke bzw. die Liniendicke von SVG-Elementen
5	**stroke-dasharray=”188.25 63.08”** Die Eigenschaft „stroke-dasharray“ löscht Teile des Rahmens oder einer Linie von SVG-Elementen. Der erste Wert bestimmt die Länge, die gezeigt werden soll. Der zweite Wert bestimmt die Länge, die gelöscht werden soll. Diese zwei Längen wechseln sich auf dem Rahmen / der Linie ab bis der Rahmen umrundet ist oder das Ende der Linie erreicht wurde. Somit ergibt sich ein gestrichelter Rahmen / Linie. Hinweis: Bei einem Kreis ist der Rahmen der Umfang. Dieser Berechnet sich aus: 2 * π * r.
6	**fill=”white”** Die Eigenschaft „fill“ bestimmt die Füllfarbe oder Schriftfarbe des SVG-Elements.
7	**transform=”rotate(135, 55, 55)”** Die Eigenschaft „transform rotate“ dreht SVG-Elemente. Die erste Zahl ist die Gradzahl und bestimmt wie weit gedreht werden soll. Die zweite und dritte Zahl bestimmen den Punkt, um welchen gedreht werden soll.
8	**id=”gauge.pointer”** Die Eigenschaft „id“ gibt einem SVG-Element eine ID, auf welche mit JavaScript zugegriffen werden soll.

© Siemens AG 2019 All rights reserved



Nr.	Code-Abschnitt aus (Tank)
1	**<rect x="10" y="10" width="30" height="10" />** Das Tag „rect“ erstellt ein Rechteck. Die Eigenschaft „x“ bestimmt den Punkt oben links des Rechtecks auf der X-Achse, „y“ auf der Y-Achse. Mit der Eigenschaft „width“ wird die Breite und mit „height“ die Höhe des Rechtecks bestimmt.
2	**<line x1="40" y1="10" x2="40" y2="110" />** Das Tag „line“ erstellt eine Linie. Die Eigenschaften „x1“ und „y1“ bestimmen den Startpunkt der Linie. Die Eigenschaften „x2“ und „y2“ bestimmen den Endpunkt der Linie.
3	**<polygon points="50,55 60,55 55,5" />** Das Tag „polygon“ erstellt ein Vieleck. Die Eigenschaft „points“ definiert die Eckpunkte des Vielecks.
4	**<text x="40" y="35" font-size="10" font-family="Helvetica, Arial, sans-serif">** **Speed** **</text>** Das Tag „text“ erstellt einen Text. Die Eigenschaft „x“ bestimmt vom unteren rechten Punkt des Textfeldes die Position auf der X-Achse, „y“ auf der Y-Achse. Die Eigenschaft „font-size“ bestimmt die Schriftgröße. Die Eigenschaft „font-family“ bestimmt die Schriftart. In diesen Tag wird der gewünschte Text geschrieben.
Nr.	**Code-Abschnitt (Aufbau eines JavaScripts)**
1	**<script>** **var newValue** **…** **document.getElementById("gauge.pointer").** **setAttribute("transform","rotate("+newValue+",55,55)")** **</script>** **„document.getElementById()“** Bestimmt das Element, auf das zugegriffen, oder verändert werden soll. **„setAttribute“** Die Funktion verändert die Eigenschaft von Elementen. Elemente die keine JavaScript Variable darstellen, werden in Anführungszeichen gesetzt. JavaScript-Variablen werden vor der Funktionsausführung durch den Variablenwert ersetzt.

Hinweis

Weitere Information zur Erstellung von SVG-Elementen finden Sie unter:

http://www.w3schools.com/html/html5_svg.asp
http://www.w3schools.com/graphics/svg_intro.asp
http://wiki.selfhtml.org/wiki/SVG

© Siemens AG 2019 All rights reserved

14 S7-Diagnose und Ladeindikator

14.1 Automatisierungsaufgabe

Der S7-Webserver bietet auf der Web-Standardseite bereits die Möglichkeit den Diagnosepuffer der CPU auszulesen.

In der Praxis ist eine Anforderung denkbar, den Diagnosepuffer in Verbindung mit anderen Informationen auf einer Anwenderseite darzustellen.

Es soll eine Anwenderseite programmiert werden, die den Diagnosepuffer der S7-Steuerung anzeigt. Den Ladevorgang soll ein animiertes Icon als Ladeindikator anzeigen.

Anforderungen an die Automatisierungsaufgabe

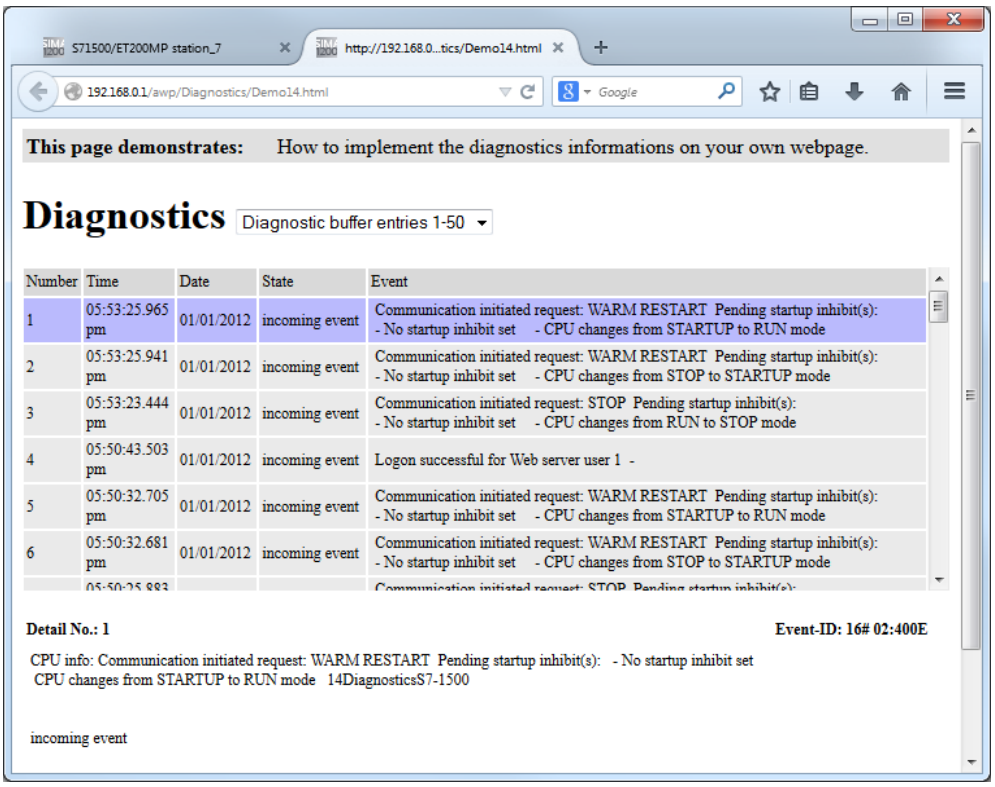
- Anzeigen von Diagnoseinformationen der S7-Steuerung auf einer Anwenderseite
- Einbinden eines Ladeindikators mit dem S7-Framework

14.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Auf der Anwenderseite werden die Diagnoseinformationen der CPU angezeigt. Beim Klick auf eine Diagnosemeldung, werden Detailinformationen zu dieser Meldung unterhalb der Tabelle angezeigt.

Abbildung 14-1: Anwenderseite mit Diagnosedaten

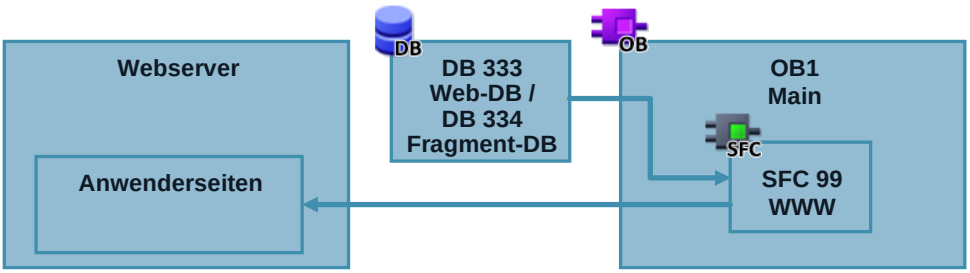


© Siemens AG 2019 All rights reserved

14.2.1 Aufbau des S7-Programms

Im S7-Programm wird ausschließlich der SFC 99 aufgerufen.

Abbildung 14-2: S7-Programm zur Anwenderseite mit Diagnosedaten und Ladeindikator



Funktionsweise des OB 1

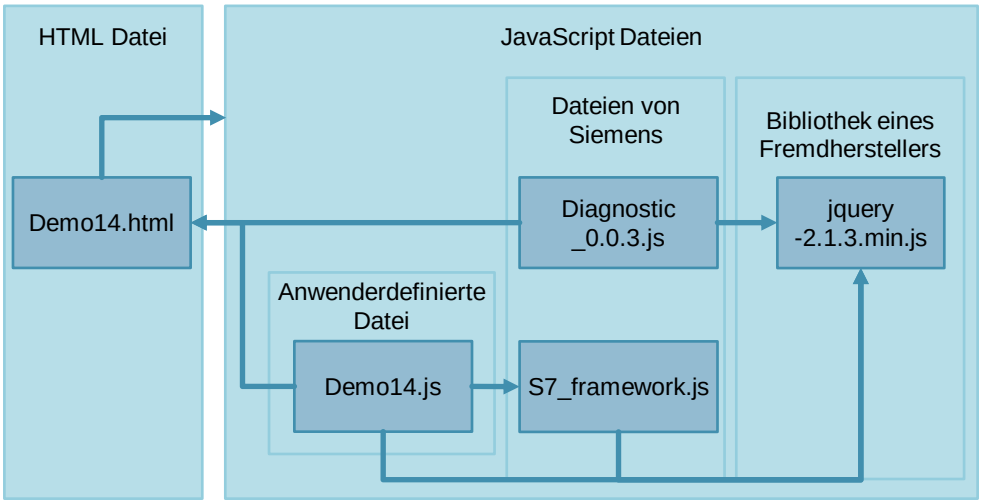
Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

14.2.2 Struktur der Web-Anwenderseite

Die grundlegende Struktur zur Anwenderseite zeigt die nachfolgende schematische Übersicht.

Aufbau der Anwenderseite

Abbildung 14-3: Schematische Übersicht der Anwenderseite



© Siemens AG 2019 All rights reserved

Demo14.html

Das HTML-Dokument definiert den Aufbau der Anwenderseite und die Aufrufe der verwendeten JavaScript-Dateien, siehe Kapitel 14.2.2.1. Im Dokument ist definiert, dass die Diagnosemeldungen per Klick in einer Tabelle angezeigt werden.

Demo14.js

Das JavaScript bindet das S7 Framework ein, um den Ladeindikator zu starten.

diagnostic_0.0.1.js

Die Datei „diagnostic_0.0.1.js“ greift auf Tags mit definierter ID zu.
In diese Tags wird die Diagnoseanzeige eingefügt.

S7_framework.js

Das JavaScript beinhaltet eine Reihe von Funktionen zur Konvertierung und Verarbeitung von Daten aus einer SIMATIC S7.
Weitere Informationen zum S7-Framework finden Sie im Kapitel 4.3

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei jQuery-Bibliothek, ermöglicht ein effizienteres Programmieren mit JavaScript. Weitere Informationen zur jQuery-Bibliothek finden Sie im Kapitel 4.3

diag.css (Stylesheet)

Stylesheet für den Ladeindikator. Dieses Stylesheet definiert die Position und die Webanimation für die Datei „…\img\sprites.png“.

S7_diagnostic.css (Stylesheet)

Stylesheet zur Gestaltung der Tabelle mit den Diagnoseinformationen

Load / Sprites (Bilder)

Der Ladeindikator ist eine animierte Datei im gif-Format, der auf die Grafikelemente einer PNG-Datei zugreift, wie nachfolgende Abbildung zeigt.

Abbildung 14-4: Bildaufbau zum Ladeindikator

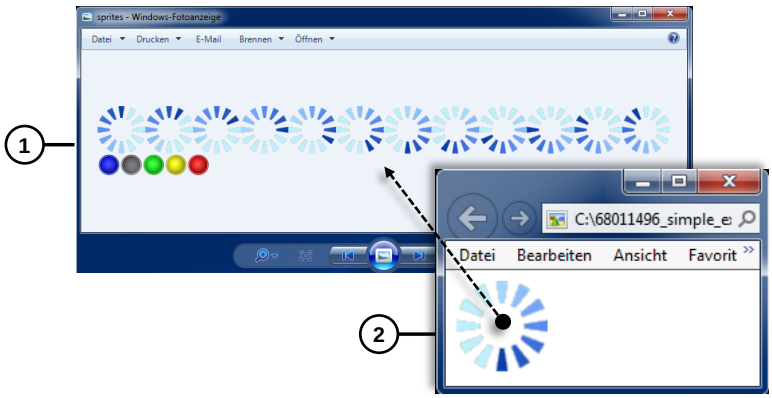


Tabelle 14-1: Schematische Übersicht der Anwenderseite

Position	Beschreibung
1	PNG-Quelldatei mit 12 Einzelbildern „sprites.png“
2	GIF-Zieldatei mit zeitlichem Wechsel der Einzelbilder (Animation) „load.gif“

© Siemens AG 2019 All rights reserved

14.2.2.1 Aufbau – Demo14.html

Nr.	Code-Beschreibung

1	Im <head> des HTML-Dokuments werden die genutzten Stylesheets und JavaScripte eingebunden:
	Stylesheets: <link rel=…>
	JavaScrips: <script type=…>
	
	`<head>`
	`  <meta http-equiv="content-type" content="text/html; charset=UTF-8">`
	`  <link rel="stylesheet" type="text/css" href="CSS/S7_diagnostic.css">`
	`  <link rel="stylesheet" type="text/css" href="CSS/diag.css">`
	`  <script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>`
	`  <script type="text/javascript" src="script/diagnostic_0.0.1.js"></script>`
	`  <script type="text/javascript" src="script/S7_framework_0.1.7.js"></script>`
	`  <script type="text/javascript" src="script/Demo14.js"></script>`
	`  <script type="text/javascript"></script>`
	`</head>`
	
	Die Eigenschaft „src“ bestimmt den jeweiligen Dateipfad.

2	In diesem „div“-Tag wird der Ladeindikator angezeigt.
	
	`<body>`
	`  <div id="loading_div">`
	`  <div class="loading_pic"></div>`
	`  </div>`

3	In diesem „div“-Tag werden die Diagnoseinformationen in Form einer Tabelle angezeigt. `<div id="diagTableDiv" width="1000px" height="250px"></div> `
4	In diesem „div“-Tag werden die Ereignisinformationen „Event“ einer Diagnosemeldung per Klick auf die jeweilige Meldung in der Tabelle gespiegelt. `<div id="diagDetailDiv" width="1000px" height="250px">` `  <span id="diagDetailNo"></span>` `  <span id="diagDetailEventID"></span> ` `  <span id="diagDetailText">DiagnosticDetail</span>` `</div>`

© Siemens AG 2019 All rights reserved

14.2.2.2 Aufbau – Demo14.js

Nr.	Code-Beschreibung

1	**Ermitteln des CPU-Typs (S7-1200 oder S7-1500)**
	Im JavaScript „Demo12.js“ wird in „$(document).ready(function(){}“ eine Funktion ausgeführt um zu ermitteln, ob eine S7-1200 oder S7-1500 verwendet wird.
	`$(document).ready(function(){`
	`  if (plcType != "1200" && plcType != "1500")`
	`  {`
	`  $.ajax({type:"GET",url:"../../../Portal/Intro.mwsl",data:"",dataType:"text"})`
	`  .done(function(webpageData){`
	`  var search12 = webpageData.search("CPU12");`
	`  var search15 = webpageData.search("CPU15");`
	`  if (search12 >= 0)`
	`  {`
	`  plcType = "1200";`
	`  $.init();`
	`  }`
	`  else if (search15 >= 0)`
	`  {`
	`  plcType = "1500";`
	`  $.init();`
	
	Alternativ haben Sie die Möglichkeit in „$(document).ready(function[]{}“ direkt die Funktionen von „$.init“ aufzurufen.
	Ersetzten Sie „plcType“ durch „1200“ oder „1500“ inklusive Anführungszeichen

2	Mit dieser Funktion wird dem S7-Framework der CPU-Typ übermittelt und die ID des „div“-Tags, in dem der Ladeindikator angezeigt werden soll.
	Das S7-Framework erkennt, wenn eine AJAX-Funktion ausgeführt wird, und blendet dementsprechend den Ladeindikator ein oder aus.
	
	`$.init = function(){`
	`  S7Framework.initialize(plcType, "#loading_div");`

	Mit dieser Funktion wird das Script aus diagnostic_0.0.3.js ausgeführt. Es fügt die Diagnoseinformationen in die „div“-Tags der HTML-Datei ein. Die „div“-Tags werden durch die entsprechenden IDs identifiziert. `diagnose.initialize(plcType);`

	Der Ladeindikator wird angezeigt, wenn das „div“-Tag mit der ID „diagTableDiv“ leer ist. Hierzu ist die folgende Abfrage programmiert.
	
	`if ($("#diagTableDiv") == "[object Object]"){`
	`  $("#loading_div").show();`
	`}else{`
	`  $("#loading_div").hide();`
	`}`

© Siemens AG 2019 All rights reserved

14.2.2.3 Aufbau – diag.css

Nr.	Code-Beschreibung

1	Das Bild „sprites.png“ für den Ladeindikator liegt im Ordner „img“.
	Die Datei „diag.css“ verweist auf das Bild sprites.png wie folgt.
	
	`.loading_pic{`
	`position: absolute;`
	`width: 64px;`
	`height: 64px;`
	`top: 50%;`
	`left: 50%;`
	`margin-top: -32px;`
	`margin-left: -32px;`
	`background-image: url(../img/sprites.png);`
	`-webkit-animation: play .8s steps(12) infinite;`
	`-moz-animation: play .8s steps(12) infinite;`
	`-ms-animation: play .8s steps(12) infinite;`
	`-o-animation: play .8s steps(12) infinite;`
	`animation: play .8s steps(12) infinite;`
	`animation-direction:reverse;`
	`}`

© Siemens AG 2019 All rights reserved

15 Wechseln von Bildern mit JavaScript

15.1 Programmieraufgabe

Es soll ein Bild, in Abhängigkeit einer JavaScript Variable, umgeschaltet werden.

Anforderungen an die Automatisierungsaufgabe

- Anzeigen von Bildern in Abhängigkeit einer JavaScript Variable

15.2 Funktionsmechanismen und Bedienung

Hinweis Zu diesem Beispiel gibt es kein S7-Programm. Die HTML-Datei finden Sie in dem Projektordner unter „html_pictureChange“

Aufbau der Anwenderseite

Auf der Anwenderseite wird ein Bild mit zwei darunter befindlichen Schaltflächen angezeigt.

Abbildung 15-1: Anwenderseite zum Bildwechsel über JavaScript-Variable

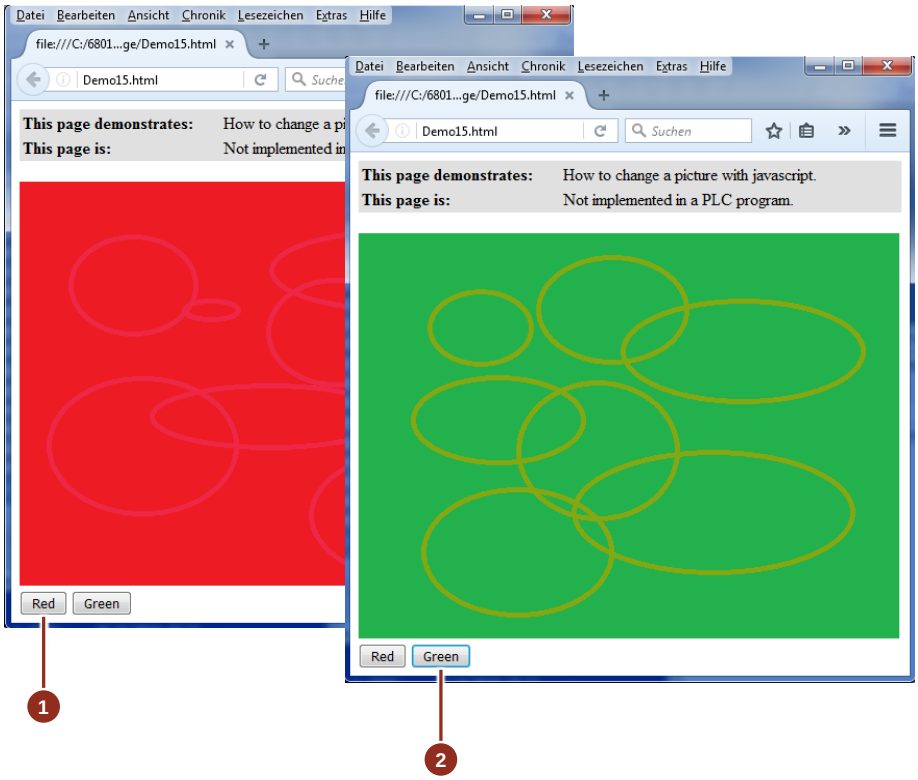


Tabelle 15-1: Handlungsanweisungen zur Bedienung der Anwenderseite

Schritt	Handlungsanweisung
1	Betätigen Sie die Schaltfläche „Red“ und das rote Bild wird angezeigt.
2	Betätigen Sie die Schaltfläche „Green“ und das grüne Bild wird angezeigt.

© Siemens AG 2019 All rights reserved

15.2.1.1 Aufbau – Demo15.html

Erstellung von SVG-Elementen

Nr.	Code-Beschreibung
1	Mit diesem HTML-Element wird beim Start der Anwenderseite das rote Bild angezeigt. `<img id="pic" src="img\red.png">`

2	Mit diesen HTML-Elementen wird eine Schaltfläche mit der Aufschrift „Red“ und eine Schaltfläche mit der Aufschrift „Green“ angezeigt.
	
	Red Green
	
	Wird die Schaltfläche „Green“ betätigt, wird die JavaScript-Funktion „changePic“ ausgelöst und der Wert „green“ an sie übergeben. Im JavaScritp beinhaltet der Parameter (color) somit den Wert (green).
	
	`<button onclick="changePic('red')">Red</button>`
	`<button onclick="changePic('green')">Green</button>`

3	Die Funktion „changePic“ ist in einem JavaScript-Anteil der HTML-Datei definiert.
	
	Um ein Bild in Abhängigkeit einer PLC-Variable zu wechseln, muss die PLC-Variable zyklisch eingelesen werden. Das zyklische Einlesen geschieht durch die if-Abfrage.
	
	Wenn der Funktion der Wert „green“ übergeben ist, wird der bisherige Pfad zum roten Bild auf den Pfad des grünen Bildes geändert.
	
	Das grüne Bild wird dadurch angezeigt.
	`<script>`
	`function changePic(color)`
	`{`
	`if (color == "green")`
	`{`
	`document.getElementById("pic").src = "img/green.png";`
	`}`
	`else`
	`{`
	`document.getElementById("pic").src = "img/red.png";`
	`}`
	`}`
	`</script>`

© Siemens AG 2019 All rights reserved

16 Schaltfläche zum CPU Neustart

16.1 Automatisierungsaufgabe

Die CPU soll über eine Schaltfläche gestoppt werden und anschließend automatisch neu starten.

Anforderungen an die Automatisierungsaufgabe

- Realisierung einer Schaltfläche zum Neustart der CPU

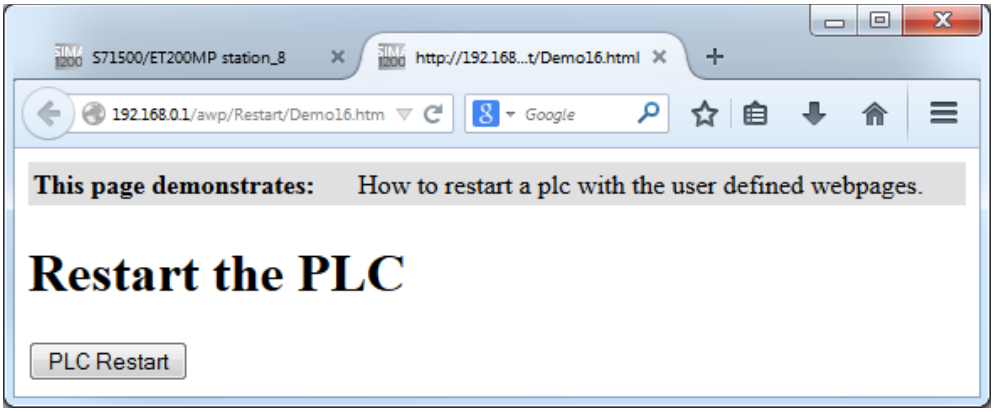
16.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Wird die Schaltfläche „PLC Restart“ betätigt, wird die CPU gestoppt und anschließend automatisch neu gestartet.

Technisch handelt es sich um einen Warmstart der CPU. Die Programbearbeitung startet neu und remanente Daten bleiben erhalten.

Abbildung 16-1: Anwenderseite mit Schaltfläche zum CPU Neustart

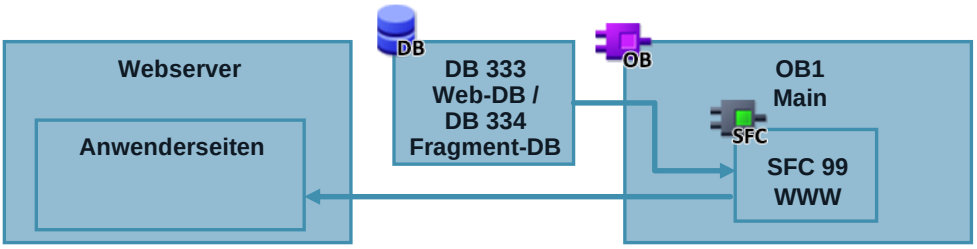


© Siemens AG 2019 All rights reserved

16.2.1 Aufbau des S7-Programms

Im S7-Programm wird ausschließlich der SFC 99 aufgerufen.

Abbildung 16-2: S7-Programm zur Anwenderseite mit Schaltfläche zum CPU Neustart



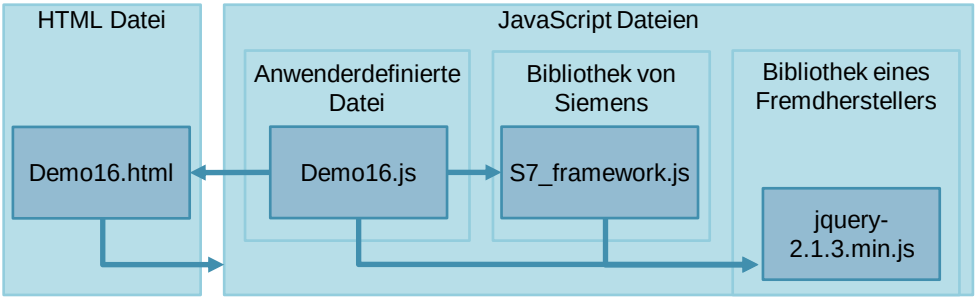
Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

16.2.2 Struktur der Web-Anwenderseite

Die grundlegende Dateistruktur der Anwenderseite zeigt die nachfolgende schematische Übersicht.

Abbildung 16-3: Schematische Übersicht der Anwenderseite



Demo16.html

In der HTML Datei „Demo16.html“ werden alle JavaScript Dateien eingebunden und die Schaltfläche definiert.

Demo16.js

Die JavaScript Datei führt den Neustart mit Hilfe des „S7_framework.js“ aus.

S7_framework.js

Das JavaScript beinhaltet eine Reihe von Funktionen zur Konvertierung und Verarbeitung von Daten aus einer SIMATIC S7.
Weitere Informationen zum S7-Framework finden Sie im Kapitel 4.3

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei jQuery-Bibliothek, ermöglicht ein effizienteres Programmieren mit JavaScript. Weitere Informationen zur jQuery-Bibliothek finden Sie im Kapitel 4.3

© Siemens AG 2019 All rights reserved

16.2.2.1 Aufbau - Demo16.html

Nr.	Code-Beschreibung

1	Im <head> des HTML-Dokuments werden die genutzten JavaScripte eingebunden:
	
	`<head>`
	`<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
	`<script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>`
	`<script type="text/javascript" src="script/S7_framework_0.1.7.js"></script>`
	`<script type="text/javascript" src="script/Demo16.js"></script>`
	`</head>`
	
	Die Eigenschaft „src“ bestimmt den jeweiligen Dateipfad.

2	Erstellung einer Schaltfläche mit dem Namen „PLC Restart“.
	
	PLC Restart
	
	`<div class="container">`
	`  <h1><span>Restart the PLC</span></h1>`
	`  <button id="restart">PLC Restart</button>`
	`</div>`

16.2.2.2 Aufbau - Demo16.js

Nr.	Code-Beschreibung

1	**Ermitteln des CPU-Typs (S7-1200 oder S7-1500)**
	Im JavaScript „Demo12.js“ wird in „$(document).ready(function(){}“ eine Funktion ausgeführt um zu ermitteln, ob eine S7-1200 oder S7-1500 verwendet wird.
	`$(document).ready(function(){`
	`  if (plcType != "1200" && plcType != "1500")`
	`  {`
	`  $.ajax({type:"GET",url:"../../../Portal/Intro.mwsl",data:"",dataType:"text"})`
	`  .done(function(webpageData){`
	`  var search12 = webpageData.search("CPU12");`
	`  var search15 = webpageData.search("CPU15");`
	`  if (search12 >= 0)`
	`  {`
	`  plcType = "1200";`
	`  $.init();`
	`  }`
	`  else if (search15 >= 0)`
	`  {`
	`  plcType = "1500";`
	`  $.init();`
	
	Alternativ haben Sie die Möglichkeit in „$(document).ready(function[]{}“ direkt die Funktionen von „$.init“ aufzurufen.
	Ersetzten Sie „plcType“ durch „1200“ oder „1500“ inklusive Anführungszeichen

2	Mit der Funktion „S7Framework.initialize(plcType,““)“ wird dem S7-Framework der CPU-Typ übergeben.
	Wird die Schaltfläche mit der ID „restart“ betätigt, wird die CPU über die Funktion „S7Framework.restartCPU();“ neu gestartet.
	
	`$.init = function(){`
	`  S7Framework.initialize(plcType, "");`
	`  $("#restart").click(function(){`
	`  S7Framework.restartCPU();`
	`  });`
	`}`

© Siemens AG 2019 All rights reserved

17 Login auf Anwenderseiten

17.1 Automatisierungsaufgabe

Die Anwenderseiten sollen über ein eigenes Login-Fenster angezeigt werden.

Anforderungen an die Automatisierungsaufgabe

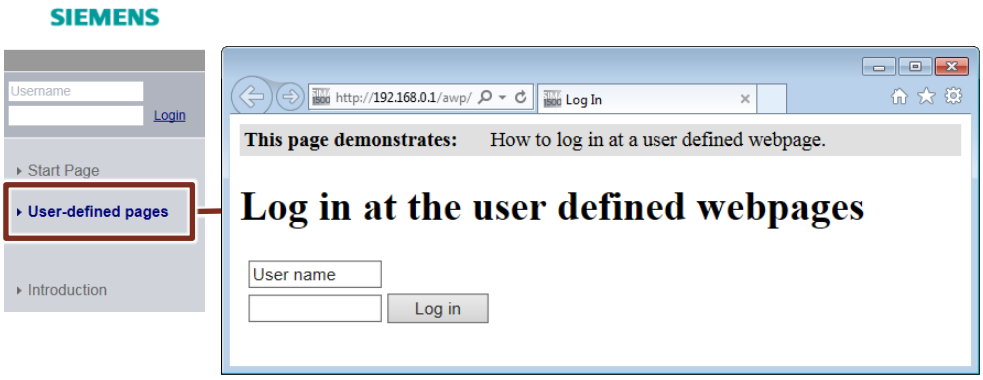
- Einbinden des Login-Fensters
- Bearbeitung der Style-Eigenschaften des Login-Fensters

17.2 Funktionsmechanismen und Bedienung

Aufbau der Anwenderseite

Über die Eingabefelder und der Schaltfläche „Log in“ melden Sie sich auf der Anwenderseite an. Hierbei gelten die Anmeldeinformationen unter Kapitel 4.1.

Abbildung 17-1: Login auf Anwenderseiten

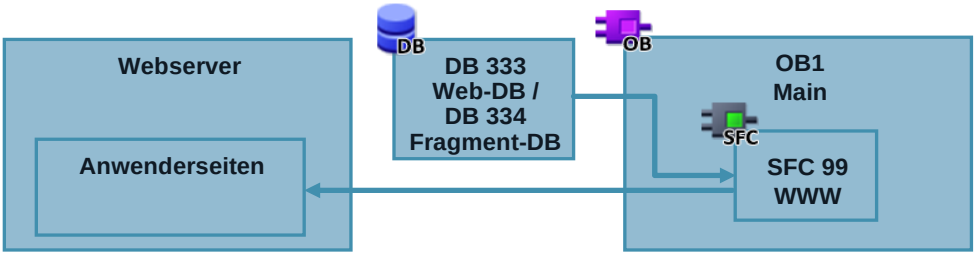


© Siemens AG 2019 All rights reserved

17.2.1 Aufbau des S7-Programms

Im S7-Programm wird ausschließlich der SFC 99 aufgerufen.

Abbildung 17-2: S7-Programm zum Login auf der Anwenderseite



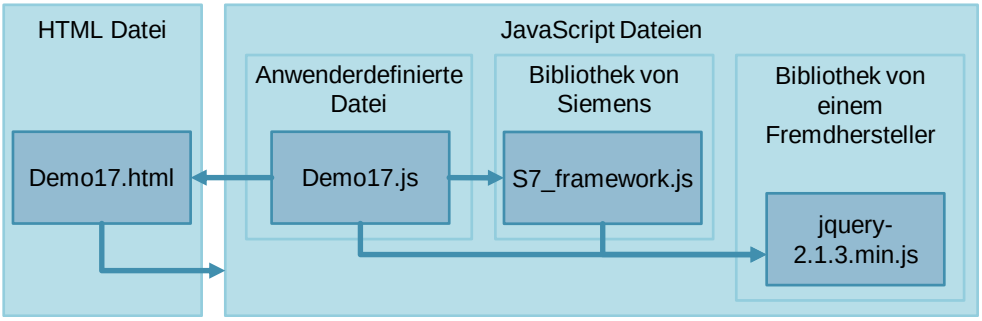
Funktionsweise des OB 1

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.

17.2.2 Struktur der Anwenderseite

Die grundlegende Dateistruktur der Anwenderseite zeigt die nachfolgende schematische Übersicht.

Abbildung 17-3: Schematische Übersicht der Anwenderseite



Demo17.html

In der HTML Datei „Demo17.html“ werden alle JavaScript Dateien eingebunden.

Demo17.js

Die JavaScript Datei ermöglicht das Ein- und Ausloggen mit Hilfe des „S7_framework.js“.

S7_framework.js

Das JavaScript beinhaltet eine Reihe von Funktionen zur Konvertierung und Verarbeitung von Daten aus einer SIMATIC S7.
Weitere Informationen zum S7-Framework finden Sie im Kapitel 4.3

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei jQuery-Bibliothek, ermöglicht ein effizienteres Programmieren mit JavaScript. Weitere Informationen zur jQuery-Bibliothek finden Sie im Kapitel 4.3

© Siemens AG 2019 All rights reserved

17.2.2.1 Aufbau - Demo17.html

Nr.	Code-Beschreibung

1	Im <head> des HTML-Dokuments werden die genutzten JavaScripte mit der dargestellten Syntax eingebunden:
	
	`<head>`
	`  <title>Log In</title>`
	`  <meta charset="UTF-8">`
	`  <script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>`
	`  <script type="text/javascript" src="script/S7_framework_0.1.7.js"></script>`
	`  <script type="text/javascript" src="script/Demo17.js"></script>`
	`</head>`
	
	Die Eigenschaft „src“ bestimmt den jeweiligen Dateipfad

2	Die „div“-Box „login“ definiert den Login Bereich als Inlineframe mit der Quelle „src=…“ aus der Web-Standardseite.
	In die „div“-Box „loginBox“ wird das Login-Fenster, durch das JavaScript, eingefügt.
	
	`<h1>Log in at the user defined webpages</h1>`
	`<div id="login">`
	`  <iframe id="WebserverIFrame"`
	`  style="display:none"`
	`  name="WebserverIFrame"`
	`  src="/Portal/Portal.mwsl">`
	`  </iframe>`
	`  <div id="loginBox"></div>`
	`</div>`

3	Im <style>-Tag wird auf die jeweiligen IDs, oder auf die Klasse zugegriffen, um die Style-Eigenschaften der Eingabefelder und Schaltflächen zu ändern.
	
	`<style>`
	`  /*S7-1500 */`
	
	`  #Login_Area_Name_InputTag{  /*Input field username*/`
	`  width:100px;`
	`  }`
	`  #Login_Area_PW_InputTag{  /*Input field password*/`
	`  width:100px;`
	`  }`
	`  #Login_Area_SubmitButton{  /*Button for login*/`
	`  width:80px;`
	`  }`
	`  .Logout_Button{  /*Button for logout*/`
	`  width: 100px;`
	`  }`
	`  #logout_user_name{  /*div of username after login*/`
	`  font-family: Arial,Helvetica Neue,Helvetica,sans-serif;`
	`  }`

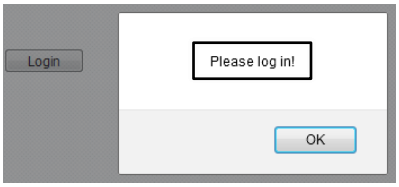
© Siemens AG 2019 All rights reserved

17.2.2.2 Aufbau - Demo17.js

Nr.	Code-Beschreibung

1	**Ermitteln des CPU-Typs (S7-1200 oder S7-1500)**
	Im JavaScript „Demo12.js“ wird in „$(document).ready(function(){}“ eine Funktion ausgeführt um zu ermitteln, ob eine S7-1200 oder S7-1500 verwendet wird.
	`$(document).ready(function(){`
	`  if (plcType != "1200" && plcType != "1500")`
	`  {`
	`  $.ajax({type:"GET",url:"../../../Portal/Intro.mwsl",data:"",dataType:"text"})`
	`  .done(function(webpageData){`
	`  var search12 = webpageData.search("CPU12");`
	`  var search15 = webpageData.search("CPU15");`
	`  if (search12 >= 0)`
	`  {`
	`  plcType = "1200";`
	`  $.init();`
	`  }`
	`  else if (search15 >= 0)`
	`  {`
	`  plcType = "1500";`
	`  $.init();`
	
	Alternativ haben Sie die Möglichkeit in „$(document).ready(function[]{}“ direkt die Funktionen von „$.init“ aufzurufen.
	Ersetzten Sie „plcType“ durch „1200“ oder „1500“ inklusive Anführungszeichen

2	Mit dieser Funktion wird dem S7-Framework der CPU-Typ übergeben.
	`$.init = function(){`
	`  S7Framework.initialize(plcType, "");`
	`}`

3	Sobald die Seite geladen ist, wird geprüft ob der Nutzer bereits über die Web-Standardseite, oder ein vorhergehenden Login, angemeldet ist.
	Ist kein Benutzer angemeldet, erscheint ein PopUp-Fenster mit dem Hinweis: „Please log in!“.
	`$(window).load(function(){`
	`  if(S7Framework.loginCheck()){`
	`  }`
	`  else{`
	`  alert("Please log in!");`
	`  }`
	`});`
	 

© Siemens AG 2019 All rights reserved

18 Performante Kommunikation via String

Beim Datenaustausch zwischen einem Webserver und einem Webbrowser werden Variablen einzeln übertragen. Möchten Sie einen umfangreichen Datenaustausch performant zwischen dem Webserver und der Steuerung realisieren, erreichen Sie dies durch die Kommunikation eines zusammengefassten Strings. Als weitere Maßnahme zur Steigerung der Performance, verwenden Sie AJAX.

AJAX

AJAX „Asynchronous JavaScript and XML“ stellt ein Verfahren zur asynchronen Datenübertragung, zwischen einem Webbrowser und dem Webserver, bereit. Der Datenaustausch findet statt, ohne die Webseite selbst neu zu laden.

18.1 Automatisierungsaufgabe

Es sollen größere Datenmengen performant an die Steuerung übertragen werden.

Anforderungen an die Automatisierungsaufgabe

- Die Werte in der Steuerung werden in einen ASCII-String mit max. 255 Zeichen konvertiert.
- ASCII-Strings werden mittels des S7-Frameworks (JavaScript) zyklisch gelesen und in vereinzelten Variablen unterteilt.
- Die Werte werden einzeln an die Steuerung geschickt.
- Die Werte werden aus einem Formular an die Steuerung geschickt.

Voraussetzung zur Kommunikation zwischen STEP 7 und der Anwenderseite mit AJAX

Auf der Anwenderseite ist das Aktualisieren und Beschreiben von PLC-Variablen mit AJAX realisiert. Hierzu muss in den Eigenschaften der Anwenderseite das Dateiformat „.json“ unter "Files with dynamic content:" ("Dateien mit dynamischem Inhalt:") hinzugefügt werden.

Abbildung 18-1: Voraussetzung - Aktualisierung über AJAX mit json unter S7

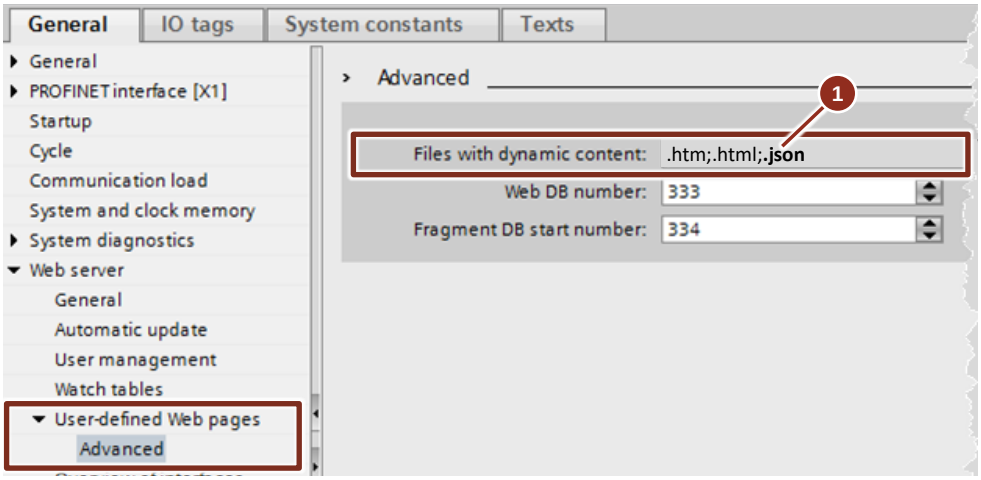


Tabelle 18-1: Handlungsanweisungen zur Erweiterung um ein Dateiformat

Schritt	Handlungsanweisung
1	Fügen Sie nach „.htm;.html;“ die Dateiendung „.json“ in das Eingabefeld ein.

© Siemens AG 2019 All rights reserved

18.2 Funktionsmechanismen und Bedienung

Auf der Anwenderseite wird ein Paket (graues Rechteck) simuliert, das sich innerhalb des schwarz umrandeten Anzeigebereichs von links nach rechts bewegt. Die Geschwindigkeit des Pakets wird über die Eingabe „Speed“ gesteuert.

Erreicht das Paket den Sensor (grüner Kreis), schaltet dieser von „Grün“ auf „Rot“ um. Der Erfassungsbereich des Sensors wird durch die schwarze senkrechte Linie markiert, wobei unterhalb dieser Linie ein Zählerwert die erfassten Pakete anzeigt.

Die Eingabe für Name und Alter wird durch die Schaltfläche [send] an die Steuerung übertragen.

Abbildung 18-2: Anwenderseite zur performanten Datenübertragung mit AJAX

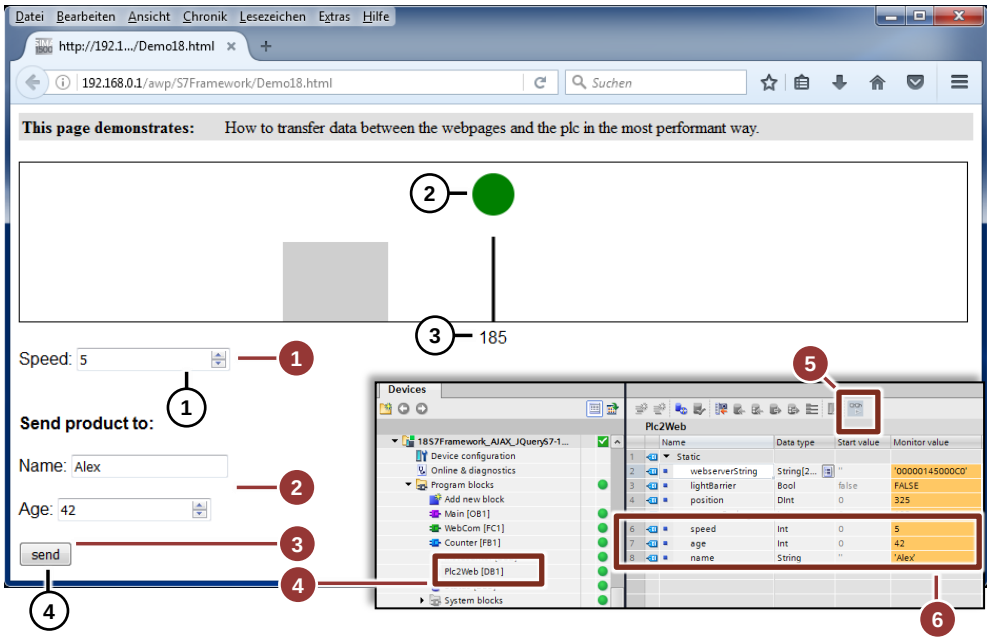


Tabelle 18-2: Anwenderseite zur performanten Datenübertragung mit AJAX

Position	Beschreibung
1	Mit dem Wert von "Speed" beeinflussen Sie die Geschwindigkeit des Pakets.
2	Signalzustand des Sensors: (Grün = frei; Rot = Paket erkannt).
3	Zählerwert zur Anzeige der bereits vorbeigefahrenen Pakete.
4	Mit [send] werden „Name“ und „Alter“ an die Steuerung übertragen.

Tabelle 18-3: Handlungsanweisungen zur Bedienung der Anwenderseite

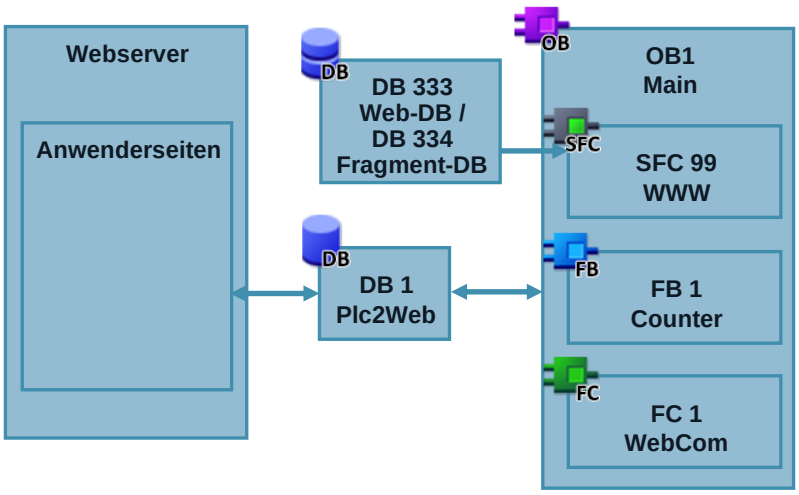
Schritt	Handlungsanweisung
1	Ändern Sie die Geschwindigkeit „Speed“.
2	Geben Sie Namen „Name“ und Alter „Age“ des Paket-Empfängers an.
3	Betätigen Sie die Schaltfläche „send“.
4	Öffnen Sie im TIA Portal den Datenbaustein „Plc2Web“.
5	Klicken Sie auf "Alle beobachten".
6	Prüfen Sie ob der Name und das Alter angekommen sind.

© Siemens AG 2019 All rights reserved

18.2.1 Aufbau des S7-Programms

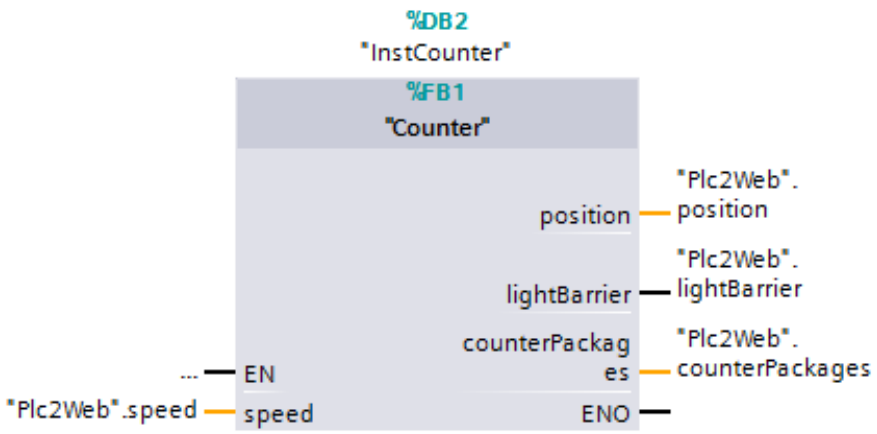
Im S7-Programm wird die WWW-Funktion aus dem OB 1 aufgerufen.
Der FB 1 „Counter“ dient der Simulation des Paketes und des Sensors.
Der FC 1 „WebCom“ dient der Konvertierung der Variablen in einen String.

Abbildung 18-3: S7-Programm zur performanten Datenübertragung mit AJAX

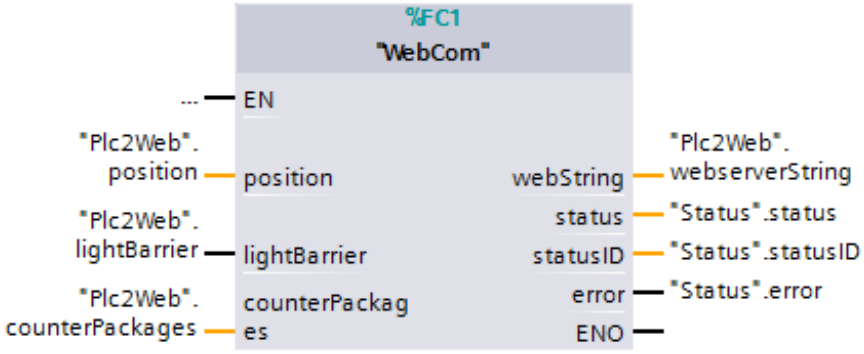


© Siemens AG 2019 All rights reserved

Funktionsweise des PLC-Programms

Nr.	Funktion
1	Der erste Bausteinaufruf im OB 1 ist die WWW-Funktion (SFC 99). Weitere Informationen zur Funktion finden Sie im Kapitel 4.2.
2	**FB 1 „Counter“ – (Simulation)** Der Funktionsbaustein berechnet die folgenden Ausgangsvariablen, als Simulationswerte für den angezeigten Ablauf in der Webseite. position: Es wird die Position des Pakets berechnet. lightBarrier: Es wird geprüft ob sich das Paket unterhalb des Sensors befindet. counterPackages: Wird ein Paket durch den Sensor erfasst, wird ein Zählerwert inkrementiert. 

© Siemens AG 2019 All rights reserved

Nr.	Funktion
3	**FC 1 „WebCom“ – (Web-Kommunikation)** Die Funktion konvertiert die Simulationswerte des FB 1 zunächst in eine Variable „webString“. Aus dieser Variablen folgt die String-Variable „webserverString“.  Möchten Sie zusätzliche Werte übertragen, müssen Sie in diesem Baustein den String erweitern.

	Die TIA Portal Systemfunktion „HTA“ konvertiert folgende Variablen-Typen in einen ASCII-String.
	• SINT
	• INT
	• LINT
	• DINT
	Bsp.:
	`// Convert DINT to ASCII-String`
	`#tempRetValHTA := HTA(IN := #position, N := 4, OUT => #tempWebString);`
	Möchten Sie Variablen vom Typ REAL oder LREAL konvertieren, beachten Sie diese zunächst in DWORD oder LWORD zu konvertieren. Diese Variablen konvertieren Sie anschließend in DINT oder LINT.

	Die Konvertierung eines Bits in einen ASCII-String erfolgt anhand einer IF-Abfrage.
	Allgemein:
	`IF bool THEN`
	`  char := '1';`
	`ELSE`
	`  char := '0';`
	`END_IF;`
	Bsp.:
	`// Convert BOOL to ASCII-String`
	`IF #lightBarrier THEN`
	`  #tempVal := #TRUE;`
	`ELSE`
	`  #tempVal := #FALSE;`
	`END_IF;`

	Die Funktion „CONCAT“ fasst zwei ASCII-Strings zusammen.
	`// Add variable 1 to variable 2`
	`#tempWebString := CONCAT(IN1 := #tempWebString, IN2 := #tempVal);`
	Bsp.:
	`#tempWebString` '000002AE0029E'
	`CONCAT` '000002AE0029E'
	`#tempWebString` '000002AE0'
	`#tempVal` '029E'

© Siemens AG 2019 All rights reserved

Nr.	Funktion
4	**DB 1 „Plc2Web“.** Im DB1 befinden sich alle Variablen, die von der CPU an die Anwenderseite gesendet, oder gelesen werden. Auch die zusammengesetzte String-Variable „webserverString“ ist in dem Datenbaustein abgelegt. **Plc2Web**

	Name	Data type	Start value	Monitor value
1	Static			
2	webserverString	String[254]	''	'000002AE0029E'
3	lightBarrier	Bool	false	FALSE
4	position	DInt	0	686
5	counterPackages	Int	0	670
6	speed	Int	0	0
7	age	Int	0	42
8	name	String	''	'Alex'

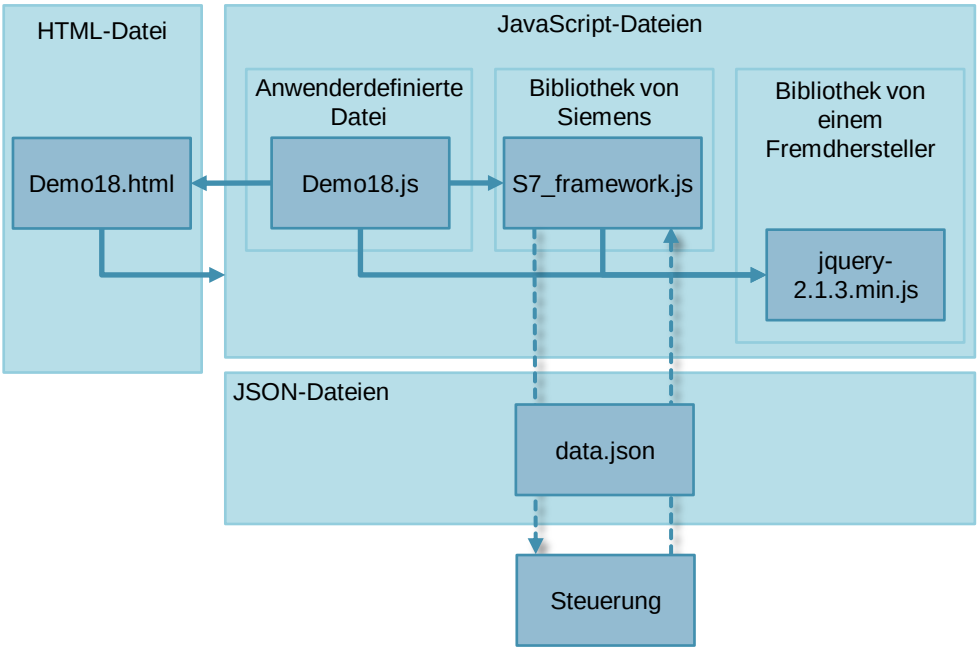
18.2.2 Struktur der Web-Anwenderseite

Die grundlegende Dateistruktur der Anwenderseite zeigt die nachfolgende schematische Übersicht.

Diese Beschreibung legt den Fokus auf die Code-Elemente des JavaScripts, wodurch grundlegende HTML-Kenntnisse für dieses Anwendungsbeispiel vorausgesetzt sind.

Aufbau der Anwenderseite

Abbildung 18-4: Schematische Übersicht der Anwenderseite



© Siemens AG 2019 All rights reserved

Demo18.html

In der HTML Datei „Demo18.html“ werden alle JavaScript Dateien eingebunden.

Demo18.js

Die JavaScript Datei „Demo18.js“ steuert den Lese- und Schreibvorgang von/auf die Steuerung, indem sie Funktionen aus der Bibliothek „S7_framework.js“ aufruft.

S7_framework.js

Das JavaScript beinhaltet eine Reihe von Funktionen zur Konvertierung und Verarbeitung von Daten aus einer SIMATIC S7.
Weitere Informationen zum S7-Framework finden Sie im Kapitel 4.3

In Demo18.js werden Funktionen des S7-Frameworks verwendet um Daten über die JSON Dateien zu übertragen.

data.json (JSON)

Die Datei dient zum Einlesen und Auslesen des zusammengesetzten Strings, über die Funktionen innerhalb des S7-Frameworks.

jquery-2.1.3.min.js (jQuery-Bibliothek)

Die Datei jQuery-Bibliothek, ermöglicht ein effizienteres Programmieren mit JavaScript. Weitere Informationen zur jQuery-Bibliothek finden Sie im Kapitel 4.3

© Siemens AG 2019 All rights reserved

18.2.2.1 Aufbau – Demo18.html

Nr.	Code-Beschreibung

1	Im <head> des HTML-Dokuments werden die genutzten JavaScripte mit der dargestellten Syntax eingebunden:
	
	`<head>`
	`  <meta http-equiv="content-type" content="text/html; charset=UTF-8">`
	`  <script type="text/javascript" src="script/jquery-2.1.3.min.js"></script>`
	`  <script type="text/javascript" src="script/S7_framework_0.1.7.js"></script>`
	`  <script type="text/javascript" src="script/Demo18.js"></script>`
	`  <script type="text/javascript" type="text/javascript"></script>`
	`</head>`
	Die Eigenschaft „src“ bestimmt den jeweiligen Dateipfad.

18.2.2.2 Aufbau – Demo18.js

Nr.	Code-Beschreibung

1	**Ermitteln des CPU-Typs (S7-1200 oder S7-1500)**
	Im JavaScript „Demo18.js“ wird in „$(document).ready(function(){}“ eine Funktion ausgeführt um zu ermitteln, ob eine S7-1200 oder S7-1500 verwendet wird.
	`$(document).ready(function(){`
	`  if (plcType != "1200" && plcType != "1500")`
	`  {`
	`  $.ajax({type:"GET",url:"../../../Portal/Intro.mwsl",data:"",dataType:"text"})`
	`  .done(function(webpageData){`
	`  var search12 = webpageData.search("CPU12");`
	`  var search15 = webpageData.search("CPU15");`
	`  if (search12 >= 0)`
	`  {`
	`  plcType = "1200";`
	`  $.init();`
	`  }`
	`  else if (search15 >= 0)`
	`  {`
	`  plcType = "1500";`
	`  $.init();`
	
	Alternativ haben Sie die Möglichkeit in „$(document).ready(function[]{}“ direkt die Funktionen von „$.init“ aufzurufen.
	Ersetzten Sie „plcType“ durch „1200“ oder „1500“ inklusive Anführungszeichen

2	Mit dieser Funktion wird dem S7-Framework der CPU-Typ übergeben.
	
	`$.init = function(){`
	`  S7Framework.initialize(plcType, "");`

3	Die Funktion „S7Framework.readData“ liest die im ausgewählten json-File (data.json) angegebenen PLC-Variablen. Sobald die Funktion ausgeführt wurde, wird der angegebene Event-Handler (updateValues) ausgeführt.
	
	`S7Framework.readData("script/data.json", ...`
	`  "init read data",updateValues);`
	`...`

© Siemens AG 2019 All rights reserved

Nr.	Code-Beschreibung
4	In der Funktion „updateValues“ werden die gelesenen Werte weiter verarbeitet. An das Array „values“ werden die Werte übergeben. Erster Wert = values[0] Zweiter Wert = values[1] … usw. `function updateValues(values)`

18.2.2.3 Aufbau – data.json

Nr.	Code-Beschreibung

1	**data.json**
	Die Datei „data.json“ enthält die Informationen zum Einlesen des Strings und die Parameter des String und die AWP Kommandos der PLC-Variablen, die beschrieben werden sollen.
	
	Die Informationen zum Auslesen des Strings entsprechen denen zum Einlesen.
	
	Es ist der zu Lesende String definiert: Plc2Web.webserverString
	Es ist der Aufbau des Strings definiert: Länge; Typ; weitere Stringvariablen
	
	`<!-- AWP_In_Variable Name='"Plc2Web".speed' -->`
	`<!-- AWP_In_Variable Name='"Plc2Web".name' -->`
	`<!-- AWP_In_Variable Name='"Plc2Web".age' -->`
	`{`
	`"val" : ":="Plc2Web".webserverString:",`
	`"len" : "8;1;4",`
	`"typ" : "2;0;2",`
	`"str" : ""`
	`}`
	
	DINT (DINT) + BOOL (BOOL) + INT (INT)
	
	“val“ entspricht dem einzulesenden String.
	
	“len“ beschreibt, welche Größe „lenght“ die Variable besitzt (Variablentypen)
	1 = 1 Bit => entspricht: BOOL
	2 = 8 Bit => entspricht: BYTE
	4 = 16 Bit => entspricht: INT / WORD
	8 = 32 Bit => entspricht: DINT / DWORD / REAL
	16 = 64 Bit => entspricht: LINT / LWORD / LREAL
	
	„typ“ beschreibt den Typ der Variable (Wird im S7-Framework definiert)
	0 = BOOL;
	1 = UINT;
	2 = INT/DINT;
	3 = REAL;
	4 = LREAL;
	5 = STRING;
	
	„str“ dient zum Laden von Variablen, die nicht in den String konvertiert wurden.

© Siemens AG 2019 All rights reserved

19 Installation

19.1 Installation von Hard- und Software

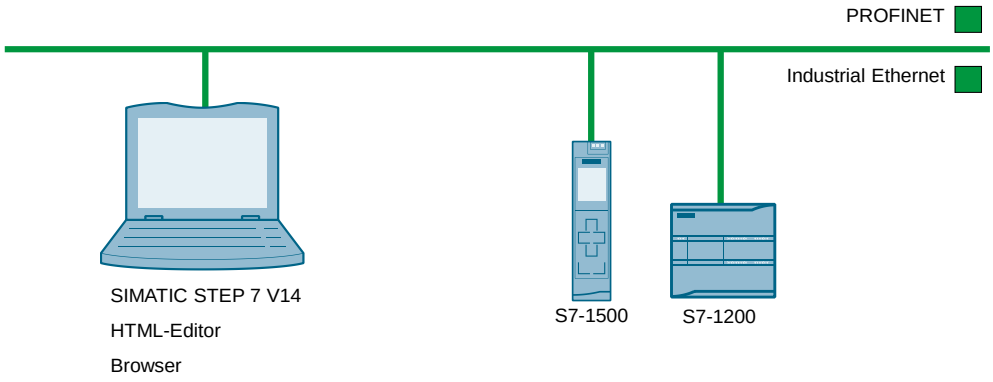
Installation der Hardware

Nachfolgendes Bild zeigt den Hardwareaufbau für die Beispiele.

Der PC (Webbrowser) ist über Industrial Ethernet mit der CPU (Webserver) über die PN-Schnittstelle verbunden.

Hierbei wurden die SIMATIC S7-Steuerungen getrennt voneinander verwendet, da in den Beispielen ausschließlich eine IP-Adresse für die S7-Steuerung verwendet wird.

Abbildung 19-1: Hardwareaufbau Beispiele



Hinweis Beachten Sie die Montage- und Anschlussrichtlinien gemäß den entsprechenden Handbüchern und Montageanleitungen.

Installation der Software

Tabelle 19-1: Installation der Softwarepakete für die Beispiele

Nr.	Aktion
1.	Installieren Sie SIMATIC STEP 7 Professional (TIA Portal).
2.	Installieren Sie auf dem PC, mit dem Sie die Anwenderseite erstellen wollen, ein Tool zur Erstellung der Anwenderseite, z. B. MS Expressions, Notepad++.
3.	Installieren Sie auf dem PC, mit dem Sie auf die Webseiten der CPU zugreifen wollen, einen Webbrowser z.B. Internet Explorer oder Firefox.

© Siemens AG 2019 All rights reserved

19.2 Installation des Anwendungsbeispiels

Tabelle 19-2: Vorgehensweise zur Verwendung der Beispiele

Nr.	Aktion	Anmerkung
1.	Entpacken Sie die Datei „68011496_examples_for_S7WebServer_CODE_v3.zip“ in Ihr Projekt-Verzeichnis.	-
2.	Starten Sie SIMATIC STEP 7.	-
3.	Öffnen Sie in SIMATIC STEP 7 das Projekt.	-
4.	Wählen sie das gewünschte Programm. Die Programme sind nach den Kapitelnummern in der Dokumentation nummeriert.	-
5.	Wechseln Sie in die Gerätesicht.	-
6.	Wenn Sie eine andere CPU verwenden möchten, tauschen Sie die CPU über den Hardwarekatalog.	-
7.	Weisen Sie in den CPU-Eigenschaften der Ethernet-Schnittstelle die IP-Adresse Ihrer CPU zu.	-
8.	Wählen Sie die CPU aus und laden Sie das gesamte Projekt in die CPU.	-
9.	Starten Sie einen Webbrowser und rufen Sie über die IP-Adresse die Webseite Ihrer CPU auf.	Weitere Informationen im Kapitel Bedienen unter den jeweiligen Beispielen

© Siemens AG 2019 All rights reserved

20 Literaturhinweise

20.1 Literaturangaben

In dieser Tabelle finden Sie eine Auswahl geeigneter Literatur.

Tabelle 20-1: Literatur

Nr.	Themengebiet
/1/	HTML und CSS, Praxisrezepte für Einsteiger Robert R. Agular mitp ISBN 978-3-8266-1779-9
/2/	HTML Handbuch Stefan Münz/Wolfgang Nefzger Franzis Verlag ISBN 3-7723-6654-6
/3/	JavaScript und Ajax, Das umfassende Handbuch Christian Wenz Galileo Press ISBN 978-3-8362-1128-4

© Siemens AG 2019 All rights reserved

20.2 Internet-Link-Angaben

In dieser Tabelle finden Sie eine Auswahl an Links zu weiterführenden Informationen.

Tabelle 20-2: Internetlinks

Nr.	Themengebiet
/1/	Siemens Industry Online Support http://support.automation.siemens.com
/2/	Link auf die Beitragsseite des Anwendungsbeispiels https://support.industry.siemens.com/cs/ww/de/view/68011496
/3/	HTML, CSS, JavaScript, http://de.selfhtml.org/ http://www.little-boxes.de/ http://www.w3schools.com
/4/	S7-1500 Webserver Funktionshandbuch https://support.industry.siemens.com/cs/ww/de/view/59193560
/5/	SVG, w3schools http://www.w3schools.com/svg/default.asp
/6/	jQuery Flot http://www.flotcharts.org/
/7/	jQuery https://jquery.com/

21 Historie

Tabelle 21-1: Historie des Dokuments

Version	Datum	Änderung
V1.0	08/2015	Erste Ausgabe
V 2.0	04/2017	Überarbeitung der folgenden Beispiele: • Sprachumschaltung • Erstellen eines Graphen Ergänzung um die folgenden Beispiele: • Erstellung von Anzeigeelementen • Wechseln von Bildern mit JavaScript • Einbindung einer Schaltfläche zum Neustart der CPU • Einbinden der Login-Möglichkeit • Performante Datenübertragung mit in Strings konvertierten Variablen
V 2.1	08/2018	Beispiel 18: Codeanpassungen
V 3.0	09/2019	Update TIA Portal V15.1

© Siemens AG 2019 All rights reserved